SISU

PUBLIKATION 97:01

RAPPORT – FEBRUARI 1997

Informationssamverkan

– trender, ansatser, möjligheter, problem

Stig Berild

SVENSKA INSTITUTET FÖR SYSTEMUTVECKLING

SISU

Innehållsförteckning

1. Inledning

Föreliggande rapport, som tagits fram inom SISUs projekt *Systemförnyelse*, diskuterar olika principer, alternativ och problem i samband med informationssamverkan. Syftet är att försöka belysa problemområdets komplexitet snarare än att föreslå lösningar. Förhoppningsvis kan synpunkterna tjäna som tankeställare för den som främst ser bekymren ligga i precisering av teknik och arkitektur.

Rapporten tar sin utgångspunkt i informationssamverkan mellan databaser och databasbaserade tillämpningar med betoning på informationsutbyte. Avsnitt 2 tar upp ett par trender som snabbt och intensivt kommer att stöpa om förutsättningar och intresse för samverkan. SQLs just nu stabila dominans inom databasområdet kommer att förbytas i en betydligt mer splittrad bild. Se vidare avsnitt 2.1. Standarder för avancerad samverkan mellan systemkomponenter har etablerats under de senaste åren. Dessa erbjuder realisering av helt nya systemstrukturer och indirekt nya förutsättningar för informationssamverkan. Se vidare avsnitt 2.2. Avsnitt 3 berör olika principiella former av informationssamverkan.

Därefter är det dags att belysa olika principer och alternativ för informationsutbyte. Avsnitt 4 diskuterar principerna medan avsnitt 5 belyser alternativ genom att i ett antal exempel ta upp olika kombinationer av förutsättningar att ta hänsyn till. Avsnitt 6 diskuterar samverkan utifrån förutsättningen att en viss tillämpning ges möjlighet att operera på ett antal databaser för att tillgodose sitt informationsbehov. Avsnitt 7 diskuterar kortfattat förutsättningar och konsekvenser av att gå över från en databasmiljö till en annan, t ex i syfte att lösa upp en splittrad databasmiljö genom att migrera tillämpningar till en och samma enhetliga miljö. Avsnitt 8, till sist, innehåller en sedvanlig avrundning.

2. Dagsläge, trender

2.1 Databasområdet

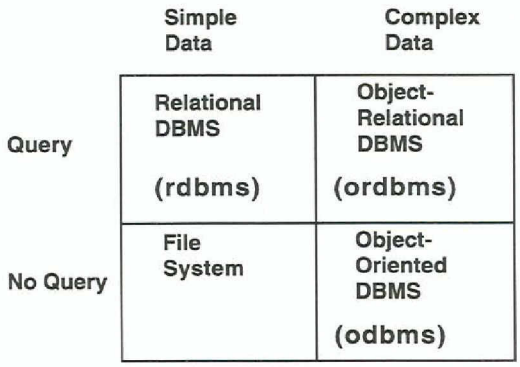
De allra flesta databastillämpningar som utvecklats under senare år utnyttjar någon form av relationsdatabashanterare (rdbms). Rdbms är med andra ord tveklöst state-of-the-art. Relationsmodellen har sin styrka i enkla datastrukturer och med användning av konventionella, standardiserade datatyper, samt inte minst dess stöd i etablerade SQL-standarder.

Allt starkare kritik och irritation har börjat riktas mot relationsmodellens begränsningar för moderna tillämpningar. Dessa behöver bl a kunna hantera

- komplexa, föränderliga datastrukturer
- funktionalitet relaterad till datastrukturen
- multimediadata; grafik, röst, video, bild, blobs, m m
- egendefinierade datatyper.

Samtidigt har det under senare år vuxit fram nya typer av databashanterare såsom objektorienterade databashanterare (odbms) och objektrelationsdatabashanterare (ordbms), den senare även under benämningen hybrid-dbms. De har marknadsförts som "nästa generation" databashanterare, något att satsa på för framtiden. Inte förvånande har marknaden börjat intressera sig mer och mer för dem även om de ännu utgör en mycket liten del av den totala databasmarknaden. Prognosinstitut brukar anse att rdbms-marknaden än så länge är en faktor femtio till hundra större än de övriga två tillsammans. Samtidigt börjar man bli överens om att den ena typen av dbms varken är mer modern eller mer gammalmodig än den andra. Varje typ av dbms har sitt berättigande för olika klasser av tillämpningar, syften och förutsättningar.

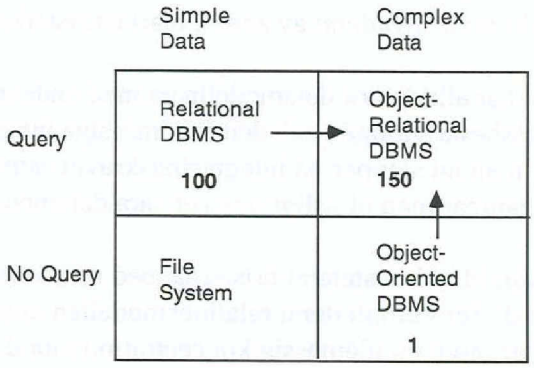
En av databasområdets förgrundsfigurer, Dr. Michael Stonebraker, brukar hänföra och karakterisera dbms till endera av fyra typer enligt följande figur:



Figur 1

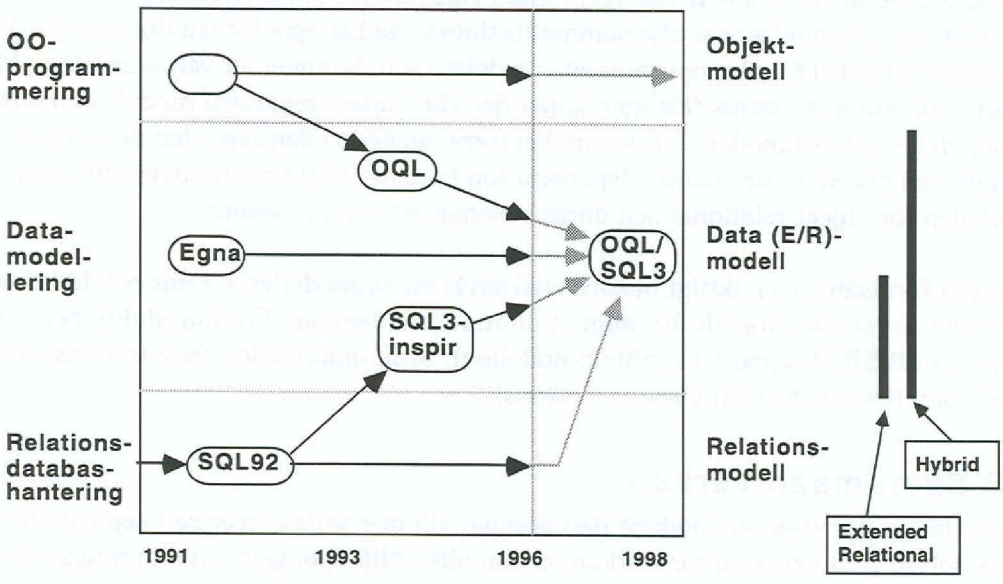
De olika typerna av dbms kommer att behöva samexistera. Verksamheter kommer i de flesta fall att ha tillämpningar baserade på dbms från olika leverantörer, baserade på olika modeller. Under en övergångsperiod kommer rdbms att dominera på grund av den redan

enorma installationsbasen. Ordbms-modellens semantiska kraft tillsammans med dess idémässiga överensstämmelse med relationsmodellen kommer att generera ett "sug" mot ordbms-lösningar framöver. Odbms-marknadens begränsade volym kommer likaledes att locka odbms-leverantörer till en omorientering mot den mer expansiva ordbms-marknaden. Aktuella standardiseringssträvanden (samarbete mellan SQL3- och ODMG-grupperna) pekar åt samma håll. De stora behoven, den stora marknaden kommer att ligga inom ordbms-rutan. Stonebraker har prognostiserat en marknadsviktning år 2005 enligt figur 2.



Figur 2

Vi får dock inte glömma att ordbms-modellen och dess gränssnitt för närvarande är ett synnerligen rörligt mål. Det kommer sannolikt att dröja minst fem år till innan rimlig stabilitet uppnås. Flera starka aktörer kommer att bevaka pågående trender och söka lösningar som ligger i linje med egna intressen. Samtidigt kommer förhoppningsvis kundkrav att bli mer välartikulerade. Det aktuella läget åskådliggörs i figur 3 (hämtad från SISU-publikation 96:11).



Figur 3

Odbms och rdbms kommer även fortsättningsvis att finna sina nischmarknader. Moderna tillämpningars behov av semantisk uttryckskraft kommer att tillgodoses av en

semantisk datamodell snarlik dem som sedan 20 år regelmässigt använts och förfinats inom konventionell datamodellering. Samtidigt ställer multimediabaserade tillämpningar krav på tillgång till nya datatyper i datamodellen samt möjlighet att skapa egna. Alltmer komplex och vittförgrenad funktionalitet ställer stora krav på anpassning, kontroll och övervakning – mekanismer och krav som delvis naturligt kan beskrivas med referens till datamodellens olika komponenter och därmed kan anses ingå i databasmodellens begreppsrepertoar.

Dbms baserade på en semantisk datamodell finns, men har levt en undanskymd tillvaro. De kommer nu att se sitt revir invaderat av andra, marknadsstarkare modellfalanger.

Objektmodell-falangen har alltid varit datamodellnära med undantag av dess inkludering av beteende i objektklassbeskrivningen och den följdmässiga integreringen mellan OO-språk och odbms. När man nu släpper på integreringskravet samt tillåter en friare syn på tolkning av beteende, hamnar man plötsligt mycket nära datamodell-falangen.

Rdbms-falangen har samtidigt konstaterat bristerna med relationsmodellen och dess tredje normalform och därför kompletterat relationsmodellen med vissa datamodell-faciliteter. Resultatet har blivit en idémässig koncentration mot datamodellhållet men färgat av tre olika falangers historia, värderingar och marknadshänsyn.

Lovvärda ansträngningar görs för att samordna och konsolidera eftersom det i ett kund-perspektiv knappast finns behov av mer än en datamodellbaserad dbms-lösning. Såtillvida har odbms-falangen representerande ODMG (en intressegrupp bestående av de flesta odbms-leverantörerna) och ISO/ANSI-organen för den officiella SQL3-standardiseringen påbörjat ett samarbete i och för framtagandet av ett enda kompro-missförslag.

Det lustiga är att datamodell-falangen, inom vars område strids-/fredsaktiviteterna utspelar sig, på grund av sitt blygsamma förflutna inte har eget förhandlingsorgan. Vilken benämning på den framkompromissade modellen som kommer att väljas är ännu oklart. Klart är däremot att rdbms-falangen kallar den för "object-relational model" och odbms-falangen för "object model". Eftersom den förra falangen i dagsläget har den största tyngden på grund av sin marknadspenetration kommer vi fortsättningsvis att kalla modellen för object-relational och dbms baserade på den för ordbms.

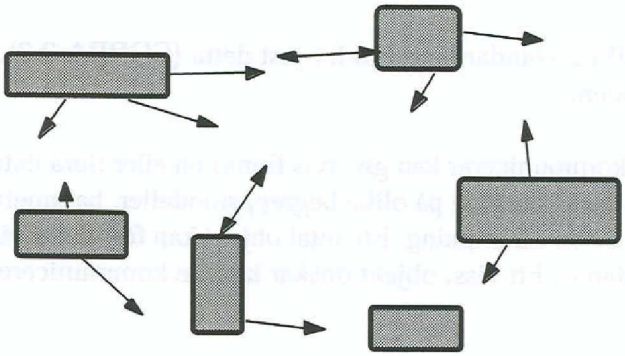
Läsaren förutsätts översiktligt bekant med såväl objektmodeller, odbms och hybrider eftersom det skulle föra allt för långt att introducera dem här. För introduktioner i ämnet hänvisas till SISU-rapport 13 (objektmodellen), SISU-publikation 96:5 (odbms) och SISU-publikation 96:11 (hybrider, ordbms).

2.2 Systemsamverkan

Aktuella trender visar att databastillämpningar allt mer sällan lever helt separata liv. Där finns behov av funktionell samverkan mellan olika tillämpningar. Tillämpningar skapas eller bryts upp som komponentstrukturer där komponenter (objekt) har ett eget ansvar och en egen funktionsrepertoar tillgänglig för andra komponenter genom meddelande-baserad samverkan. Object Request Brokers (ORBs), avancerade dokumenthanterings-miljöer, m m erbjuder infrastrukturen. ORB-marknaden är i dagsläget en kamp mellan de leverantörer som agerar under Object Management Group (OMG) å ena sidan och

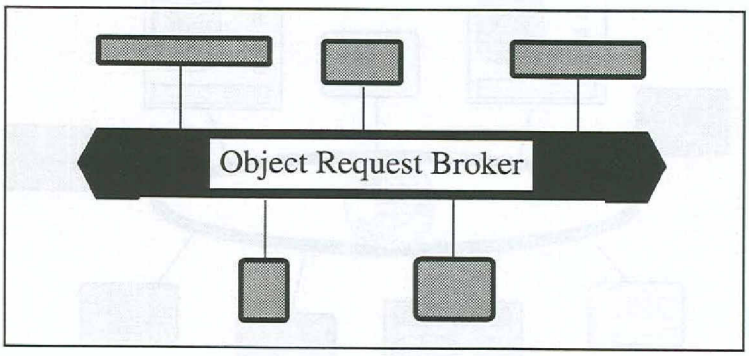
Microsoft å den andra sidan. Båda har i stort sett samma vision men med den huvudsakliga skillnaden att OMG representerar öppna lösningar medan Microsoft knappast ser just öppenheten som en intressant faktor. Låt oss bortse från marknadsspelet och istället helt kort beröra vad som ligger bakom visionen eftersom dennas realisering kommer att få mycket stark inverkan på förutsättningarna för informationssamverkan. Vi väljer att resonera utgående från OMGs perspektiv.

Syftet är att åstadkomma samverkan mellan systemkomponenter (objekt) i enlighet med ett objektorienterat synsätt. Objekten lever där "sina fria liv" och umgås genom att skicka meddelanden till varandra. Ett meddelande är i allmänhet en begäran om att få utfört någon av de operationer (methods) det andra objektet klarar av. Varje meddelande innehåller, förutom vilket objekt som anropas och vilken operation, vilka eventuella övriga uppgifter som operationen behöver känna till för att kunna utföras samt uppgift om vilket objekt som skickat meddelandet (för den händelse denne behöver informeras om utfall av operationen).



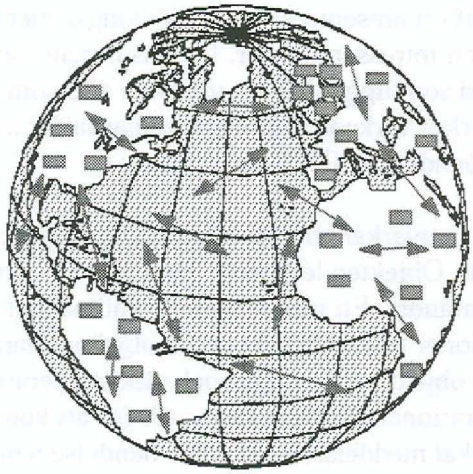
Figur 4

Var objektet befinner sig, vilken hårdvara och vilket operativsystem det opererar under, vilket programmeringsspråk det implementerats i, m m är tekniska aspekter som just ORB har till uppgift att klura ut och anpassa sig till. Logiskt kopplar objekten upp sig på en ORB (figur 5) varefter meddelandeutväxligen kan börja. I praktiken krävs förstås ett antal handgrepp för att åstadkomma detta.



Figur 5

En ORB begränsas av ett nät. Alla objekt på nätet är med andra ord nåbara. Men mycket vill ha mer. Exempelvis att kunna kommunicera med objekt i andra nät, med alla objekt man känner till och har rätt att kommunicera med oavsett var de befinner sig (figur 6).

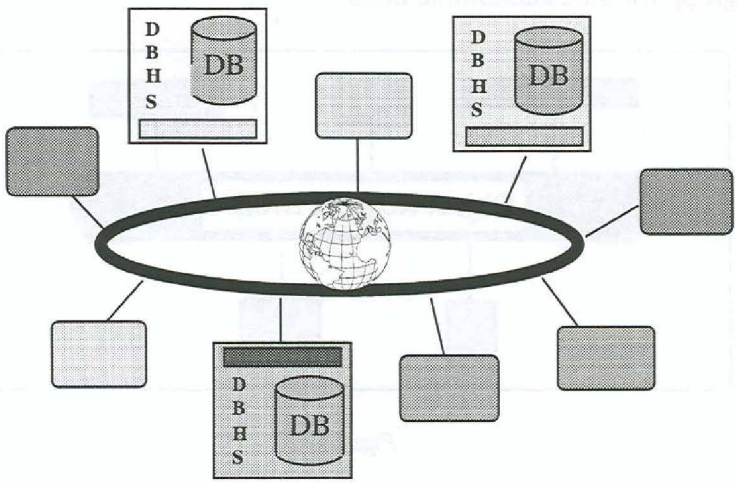


Figur 6

Kompletterande OMG-standarder erbjuder just detta (CORBA 2.0). Produkter börjar komma på marknaden.

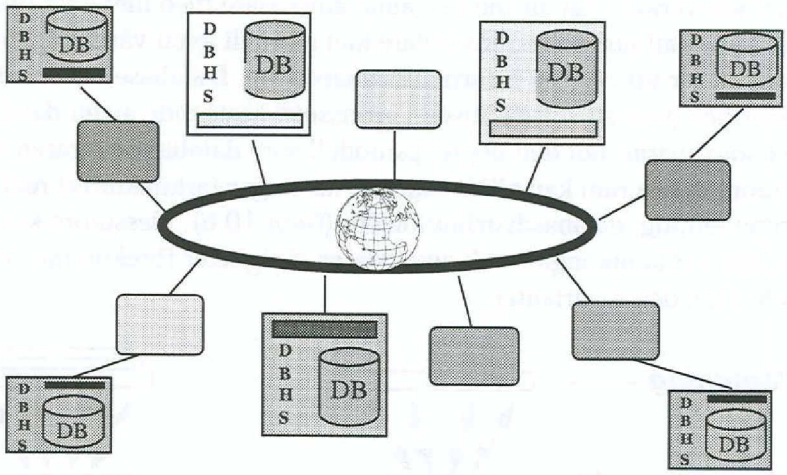
Bland objekt som kommunicerar kan givetvis finnas en eller flera databas-servrar. Se figur 7. Dessa kan vara baserade på olika begreppsmodeller, ha innehåll som svarar mot en eller flera delar av en tillämpning. Ett antal objekt kan förväntas vilja kommunicera med respektive databas. Ett visst objekt önskar kanske kommunicera med flera databaser.

Begreppsmodell:	Den modell som uttrycker vilka begrepp och samband mellan dem som behövs för att uttrycka informationsmodeller. Tillämpningsneutral. T ex: relationsmodell, Entity-Relationshipmodell, ...
Informationsmodell:	En abstraktion av en aktuell "verklighet" uttryckt i form av en modell innehållande de typer av företeelser eller objekt som finns i den aktuella "verkligheten", tillsammans med relevanta samband mellan dessa typer. Tillämpningsanpassad. T ex: Produktmodell, Orderhanteringsmodell, ...



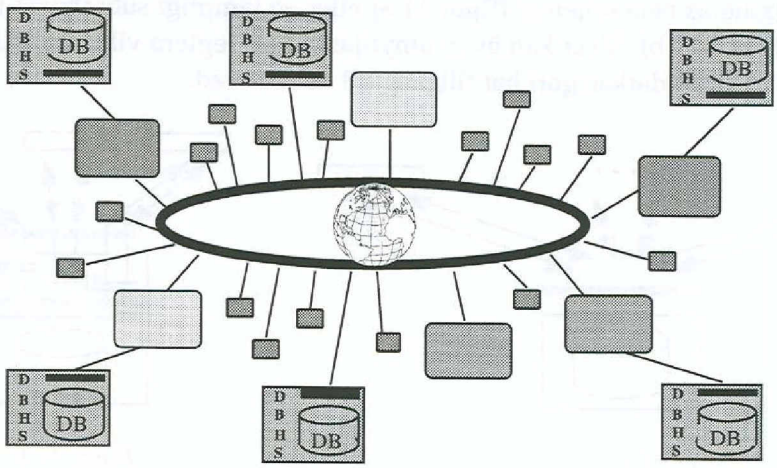
Figur 7

Därutöver kan det finnas objekt som har egenkontroll över sin "egen" databas. Andra objekt har eventuellt tillgång till dessa data men då endast genom det "ägande" objektets gränssnitt. Se figur 8.



Figur 8

En trend är att omstöpa tidigare passiva dataobjekt i vanliga databaser till aktiva, dynamiska objekt. Ett sådant aktivt objekt innehåller sina egna data men eventuellt också kontroll över diverse samband med omgivningen. Följdriktigt måste detta innebära att antalet objekt på ett globalt ORB växer lavinartat (fine grained objects) och att ORBs delvis övertar de roller konventionella dbms traditionsenligt dominerat. Se figur 9.



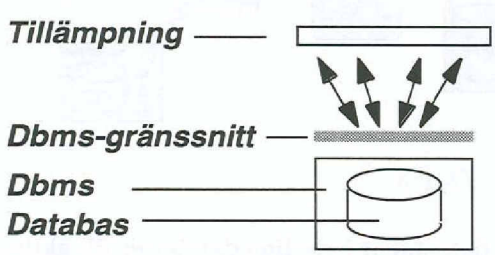
Figur 9

Lägg därtill de alltmer avancerade Internet- och Intranet-tjänsterna, som banar vägen för nya dynamiska systemsamverkansformer samt dynamiskt rörliga objekt över ett globalt ORB.

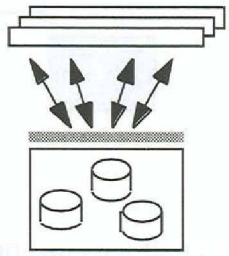
Trenderna inom systemsamverkan skapar otvivelaktigt helt nya möjligheter för informationssamverkan – men sannolikt även nya bekymmer. Nästa avsnitt diskuterar informationssamverkan ur ett generellt perspektiv varefter avsnitten 4 och 5 mer i detalj belyser olika förutsättningar för informationsutbyte med hjälp av ett antal exempel.

3. Informationssamverkan, översikt

I sin enklaste form sker samverkan mellan en databasmekanism och en tillämpning. Tillämpningen samverkar i sin tur med ett antal användare med mer eller mindre diversifierade informationsbehov. Användare kan normalt även vända sig direkt till databashanteraren för viss typ av informationshantering. Databasen tillhandahåller en informationsservice över ett standardiserat gränssnitt bestående av ett databasschema och en syntax som svarar mot den begreppsmodell som databashanteraren arbetar under (figur 10 a). Inom denna ram kan olika tekniska lösningar (arkitekturer) realiseras såsom klient/serveruppdelning, databasdistribution etc (figur 10 b). Dessutom kan olika typer av inskränkningar i tillämpningars och användares rättigheter förekomma. Vi bortser fortsättningsvis från dessa varianter.

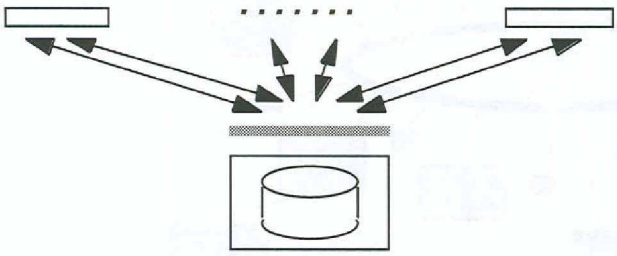


Figur 10 a

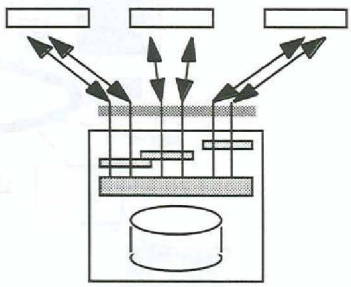


Figur 10 b

En tillämpning är en logisk gruppering av en funktionalitet för något syfte. Självfallet är det inget som hindrar att flera sådana grupperingar jobbar mot en och samma databas. Antingen exponeras hela schemat (figur 11 a) eller ett lämpligt subschema (vy) mot en tillämpning (figur 11 b). Vyer kan även utnyttjas för att reglera vilka uppgifter en användare eller användarkategori har tillgång till och för vad.



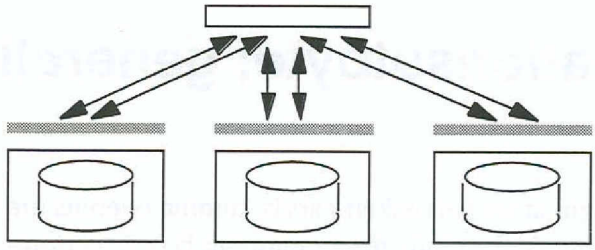
Figur 11 a



Figur 11 b

Hittillsvarande förutsättningar hanterar de flesta dbms regelmässigt. Mekanismer finns även för att transformera överförda data mellan dbms-modellens uppfattning och tillämpningsspråkets uppfattning. I rdbms-sammanhang tillämpas normalt en cursor-mekanism.

Fler och fler tillämpningar behöver vända sig till mer än en databas för att tillgodose sina informationsbehov. Baseras aktuella dbms på olika begreppsmodeller gäller förmodligen helt olika gränssnitt för varje typ av dbms. Utvecklaren behöver vara "språkbegåvad". Därutöver läggs en ansenlig konverteringsbörda på tillämpningen.

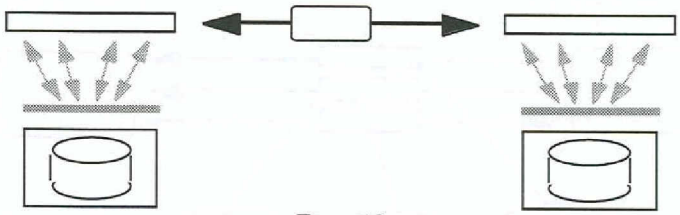


Figur 12

Förutsättningen är typisk bl a vid data warehouse-ansatser. Typiskt är också att nya tillämpningar har behov av samverkan med existerande, ofta stora så kallade "legacy systems". I avsnitt 6 diskuteras gränssnittsproblematiken vidare.

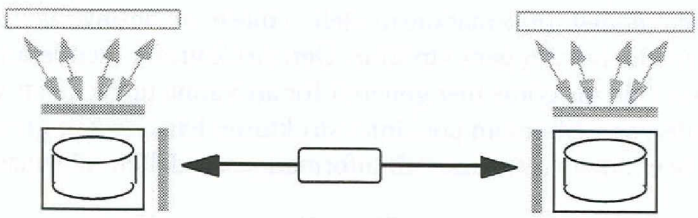
Parallellt ökar även behovet av informationsutbyte mellan tillämpningar som ett komplement till gemensamgörande via databaser. I sin enklaste form sker det genom en avgränsad överföring (sekvens av meddelanden) enligt överenskomna regler.

Utbyte mellan tillämpningar innebär att databasens uppgifter först förädlas inom avsändande tillämpning på ett sätt som bara denna känner till och har kontroll över. Den semantiska innebörden av det som skickas är således också dold under denna förädling. Detta är alldeles i sin ordning mellan parter som på någon grund etablerat ett avtal sinsemellan. Likaså är det i sin ordning om sändande tillämpning medelst lämplig, tolkbar informationsmodell talar om för omgivningen vilken information tillämpningen är villig att tillhandahålla. Hur och var tillämpningen hämtar och förädlar informationen är en privatsak så länge som utlovade leveranser och utlovad kvalitet erhålls.



Figur 13

Databaser kan även, via generella export/importmekanismer, direkt utbyta information mellan sig utan att passera viss tillämpning. Förutsättningarna är principiellt enkla eftersom avsändarens databasschema direkt kan utgöra grunden för den informations-modell som reglerar vilken möjlig information som står till buds för överföring. Givetvis är det i slutändan fortfarande fråga om en förhandling mellan sändare och mottagare.



Figur 14

Samverkan realiseras under denna förutsättning utan att de enskilda tillämpningarna eller databaserna behöver modifieras. Mer om informationsutbyte i följande avsnitt, 4 och 5.

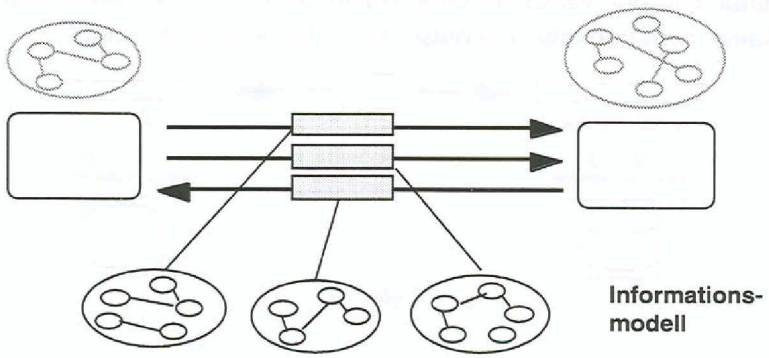
4. Informationsutbyte; generellt

4.1 Principer

Den enklaste strategin är att helt enkelt parvis komma överens om vilka typer av uppgifter som tillämpning A ska sända till tillämpning B och tvärtom, samt att för varje typ av uppgift (meddelandetyp) kontraktsskriva övriga villkor som ska gälla. Meddelandetypens struktur kan ses som en enkel form av informationsmodell svarande exakt och endast mot den aktuella meddelandetypen.

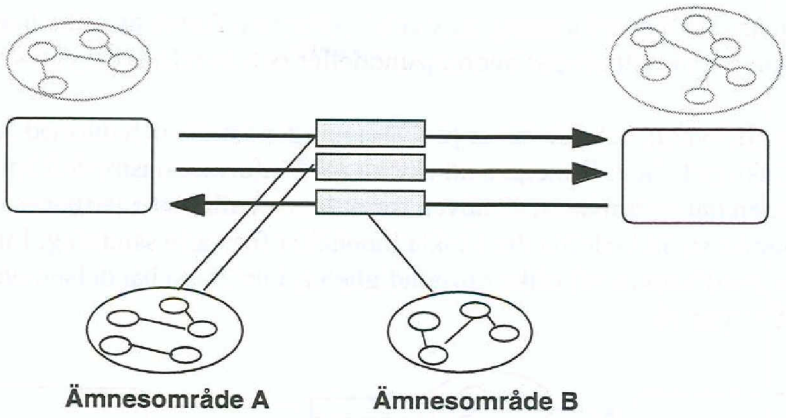
Givet dessa överenskommelser räcker det därefter för respektive part att etablera generella mekanismer för transformation mellan meddelandetypens informationsmodell och de egna interna modellerna. Arbetet kan givetvis vålla olika grad av problem beroende på använd intern datamodell samt graden av semantisk överensstämmelse med den gemensamma modellen.

Figur 15 visar denna förutsättning med bilaterala överenskommelser per meddelandetyp. Varje meddelandetyp beskrivs semantiskt genom "sin egen" informationsmodell som formulerats på något för parterna lämpligt sätt. Sannolikt svarar här meddelandets strukturella uppbyggnad mot informationsmodellen på ett entydigt sätt. Respektive tillämpnings interna informationsmodell eller databasschema indikeras i gråton över respektive tillämpningssymbol.



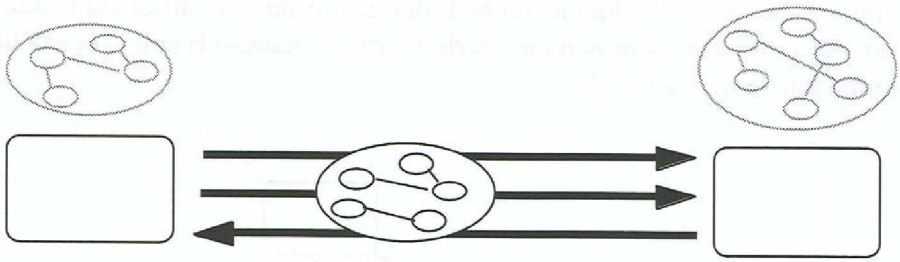
Figur 15

Om samverkan är av mer permanent art eller består av många meddelandetyper, kan det finnas anledning att generalisera informationsmodellen till att gälla t ex per ämnesområde eller per kontraktsomgång och sedan definiera valfritt antal meddelandetyper med referens till denna modell. Informationsmodellen måste nu definieras explicit och fristående från meddelandetypens struktur. Den strukturella meddelandetypsuppbyggnaden måste i sin tur göras mer generell för att kunna uttrycka ett variabelt antal olika meddelandetyper. Eftersom position i strukturen här inte kan ge någon semantisk vägledning måste explicita referenser till informationsmodellen tillgripas.



Figur 16

Mellan tillämpningar med omfattande och/eller permanent informationsutbyte kan i förlängningen komma att upprättas en generell informationsmodell som täcker alla behov av informationsutbyte dem emellan.



Figur 17

En variant på detta tema är en situation där en viss tillämpning annonserar för omvärlden att den är villig att leverera information till alla intresserade. Den information som erbjuds presenteras i en eller flera informationsmodeller, uttryckta på det sätt leverantören själv finner lämpligt. Leverantören anger även övriga principer som ska gälla under en överföringsomgång, tillgänglighet, m m. Det är knappast fråga om någon överenskommelse, snarare ett "take it or leave it". Eventuellt kan specifika avtal med respektive mottagare ändå behöva formuleras för att reglera tider, hantering av extra-ordinära omständigheter, pris, m m.

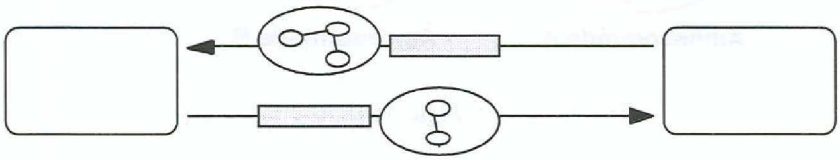
Över till ett specialfall.

Minsta förhandlingsmöda krävs om hela informationsmodellen medföljer sändningen. Logiskt kan sändningen delas upp i två separata subsändningar, där den första är informationsmodellen som data och den andra är databasinformation i enlighet med informationsmodellen.

Överföringssyntaxen liksom vanlig gränssnittssyntax baserar sig på den bakomliggande begreppsmodellen. Har både databasinformationen och informationsmodellen modellerats med hjälp av samma begreppsmodell ligger det nära till hands att också tillämpa samma syntax för överföring av såväl databasinformation som informations-modelluppgifter. I det läget räcker det med att den enda begreppsmodellen måste vara

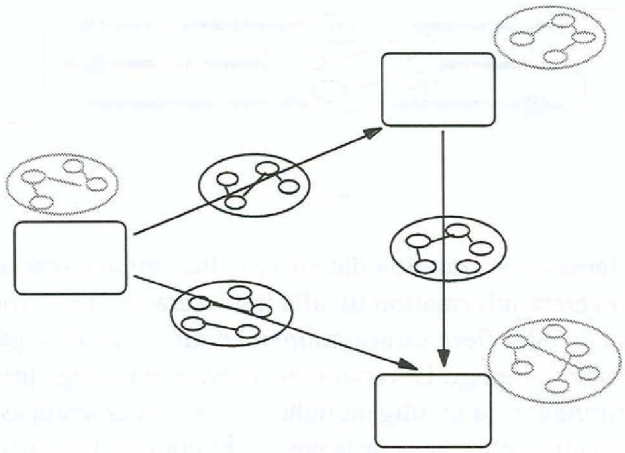
känd tillsammans med överföringssyntax och kodformat. Används olika begrepps-modeller måste två uppsättningar begreppsmodeller och överföringssyntaxer tillämpas.

Begreppsmodell, syntax och format är ju tillämpningsneutrala och därmed användbara för alla upptänkliga behov. Principen att skicka med informationsmodell kan vara intressant i lägen där informationsbehoven formulerats i allmänna termer och mottagaren överlåtit på sändaren att bedöma det exakta innehållet för varje sändning. Innehållet kan tänkas variera med hänsyn till olika omständigheter, inträffade händelser, etc hos avsändaren. Se figur 18.



Figur 18

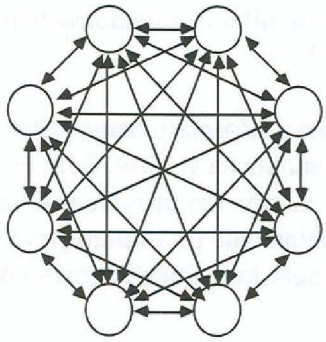
Så långt är det mesta gripbart och rimligt att administrera. Antag nu att C tillkommer. Om B är intresserad av vad A kan leverera är det inte osannolikt att också C kan vara intresserad. A får prata sig samman med både B och C. Kanske B och C också har gemensamma intressen (figur 19).



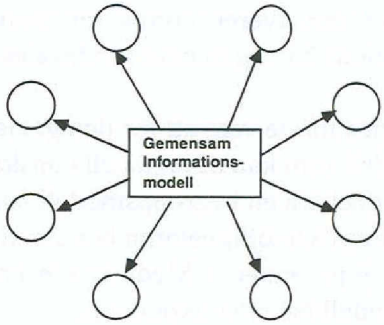
Figur 19

I det generella fallet måste varje tillämpning eller nod, förutom sin egen informations-modell, hålla reda på två externa. Kanske inte så farligt, men tillkommer så D, E, F, ... blir snart situationen enligt figur 20 a.

Sänder A liknande uppgifter till många vore det praktiskt om informationen kunde levereras med referens till samma informationsmodell för samtliga och enligt samma syntax och inkodningsformat. Om B tar emot från många skulle det likaledes gynna B om denne kunde dela upp och tolka denna inkommande information enligt samma regler oavsett avsändare. Se figur 20 b.

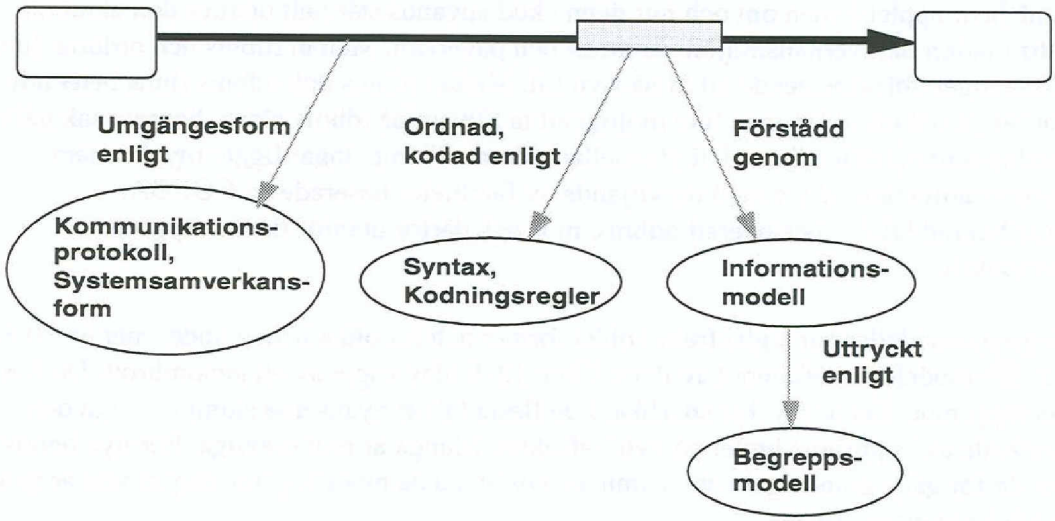


Figur 20 a



Figur 20 b

Grundidén är att ersätta en situation liknande figur 20 a med en betydligt smidigare enligt figur 20 b. I det senare fallet ankommer det på tillämpningarna att komma överens om ett gemensamt modellspråk (begreppsmodell) och att med hjälp av detta etablera en informationsmodell (schema) som svarar mot det samlade tänkta informationsutbytet. Varje specifikt informationsutbyte kan därigenom genomföras med hänvisning till lämpliga delar av schemat. Dessutom måste överenskommelse etableras avseende dels en generell överföringssyntax, dels acceptabla inkodningsprinciper.



Figur 21

En attraktivt enkel vision med en betydligt bistrare och mer mångfacetterad realitet, vilket kommer att framgå i avsnitt 5, nedan.

4.2 Gemensam begreppsmodell

Det vi eftersträvar är en möjlighet att använda samma syntax, kodningsformat och semantik för överföring mellan alla intressenter inom någon avgränsad informations-utbytesvärld, t ex inom en bransch, en verksamhet eller en verksamhets affärsområde.

Denna överenskommelse gäller endast i samband med överföring. Hur respektive tillämpning internt hanterar sin information är tillämpningens lokala ensak, likaså vilken typ av dbms som används. Dessutom genomsyras antagligen databasschemat av lokala

semantiska valörer, överenskomna internt mellan de tilltänkta databasanvändarna och som inte bedömts ha någon extern relevans.

Grundprincipen måste vara att för den gemensamma informationsmodellen tillämpa en begreppsmodell som kan uttrycka alla önskade semantiska valörer. Som en tumregel borde detta innebära en begreppsmodell som har så mycket uttryckskraft att de semantiska uttrycksmöjligheterna hos samtliga inblandade tillämpningars lokala dbms-modeller kan representeras. Medvetna inskränkningar kan givetvis göras i de fall inte en parts fulla modell behöver exponeras.

Som ovan nämnts kan en mycket klar trend mot någon form av semantisk datamodell (or-modell) skönjas både från relationsmodell-falangen och objektmodell-falangen. Or-modellen är kompatibel med relationsmodellen men påbyggd med betydligt mer uttryckskraft. Den svarar även mycket väl mot objektmodellens statiska perspektiv, d v s den del av modellen som normalt återfinns i ett odbms.

Dynamik eller objektbeteende är knappast av intresse annat än mellan grupper av odbms-tillämpningar eftersom både rdbms och ordbms baserar sig på en filosofi som klart skiljer mellan data och funktion. De data som skickas i det senare fallet kan visserligen mycket väl innehålla funktionella beskrivningar i form av kodsegment o dyl (jmf Java-applets) men om och hur denna kod används står helt utanför den aktuella informationssamverkansmiljöns kontroll och påverkan. Mellan rdbms och ordbms finns alltså inget objektbeteende att ta hänsyn till. Mellan rdbms och odbms finns beteende endast på odbms-sidan men inte möjligt att ta tillvara på rdbms-sidan. Samma sak gäller mellan ordbms och odbms. Utbyte mellan odbms-tillämpningar ligger mycket nära systemsamverkan, d v s med utnyttjande av faciliteter baserade på CORBA, distribuerad OLE, distribuerad odbms, m fl och därför utanför denna rapports perspektiv.

Begreppsmodeller har tagits fram i olika former och för olika behov under mer än 20 års tid. ER-modellen, definierad av Peter Chen 1976 blev något av ett genombrott. De flesta begreppsmodeller är mycket snarlika. I de flesta fall är nyanserna slumpmässiga och orsakade av "uppfinna hjulet på nytt"-effekter. Många är rent onödiga. För nya behov bör därför gälla grundregeln att anamma en existerande modell, gärna någon som är spridd och/eller standardiserad.

Bland mera kända begreppsmodeller återfinns

- Express tillhörigt STEP-standarden
- CDIF-modellen för definition och överföring av CASE-data mellan verktyg
- EDIFACT-modellen för definition och överföring av affärsdata (dock mer struktur än semantik)
- IDEF1X framtaget inom amerikansk flyg- och försvarsindustri
- NIAM-modellen framtagen av Dr Nijssen på 1970-talet
- IRDS-modeller för hantering av data i repositorier
- m fl.

4.3 Syntax, inkodningsprinciper

STEP-, EDIFACT- och CDIF-standarderna innehåller samtliga syntaxregler för uppbyggnad av meddelandesekvenser. Tre grundprinciper kan särskiljas:

a. Hela informationsmodellen medföljer en sändning. Kräver endast kunskap hos mottagaren om hur informationsmodellen ska tolkas, d v s begreppsmodellen måste vara känd.

b. Informationsmodellreferenser medföljer sändningen. Kräver att informationsmodellen är känd hos mottagaren. Indirekt måste givetvis även använd begreppsmodell vara känd.

c. Sändningen sker i enlighet med fördefinierat meddelandeformat. Meddelandeformatet måste vara känt hos mottagaren. Indirekt måste på något sätt mottagaren ha möjlighet att entydigt tolka innehållet.

EDIFACT tillämpar alternativ c, CDIF i huvudsak b men med möjlighet till a för speciella informationsmodelltillägg.

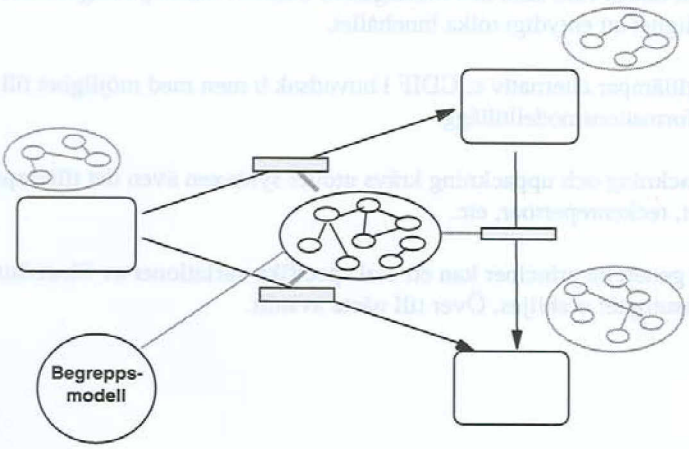
För rätt inpackning och uppackning krävs utöver syntaxen även det tillämpade kodformatet, teckenrepertoar, etc.

Inom dessa generella principer kan ett otal specifika variationer av förutsättningar för informationsutbytet urskiljas. Över till nästa avsnitt.

5. Informationsutbytesvarianter

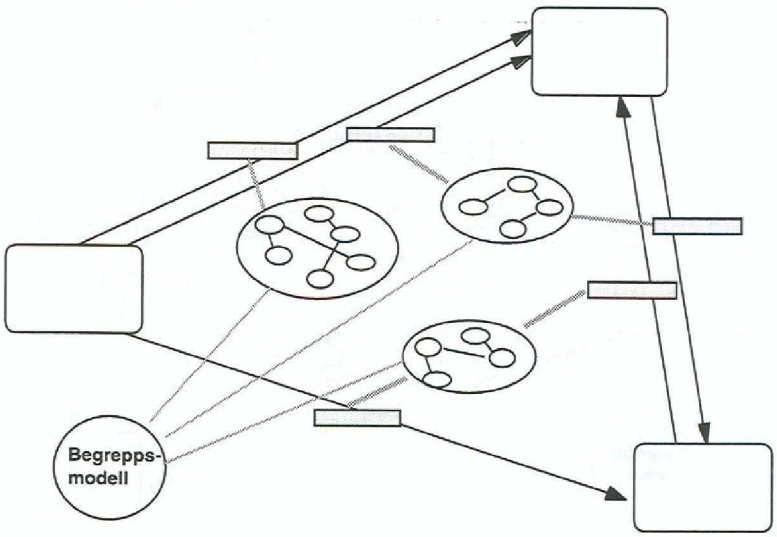
5.1 Enkla informationsutbyten, generella överenskommelser

I en idealsituation förekommer bara en enda överenskommen informationsmodell med tillhörande överföringssyntax och kodregler. Informationsmodellen täcker allt tänkbart utbyte mellan alla kombinationer av sändare och mottagare. Samtliga parter som accepterar förutsättningarna och har möjlighet att internt hantera sändning/mottagning enligt dessa förutsättningar kan här enkelt idka informationsutbyte. Vad som återstår är att definiera innehåll för en sändning med referens till informationsmodellen. Se figur 22 där streckade modeller symboliserar respektive parts interna modell.



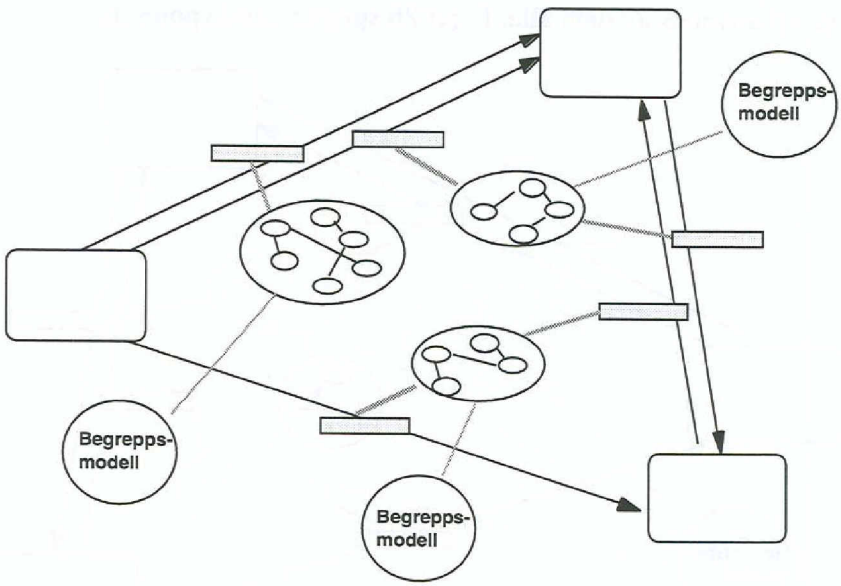
Figur 22

Att samlas kring en gemensam informationsmodell kan kanske låta sig göras med få parter inblandade, om ens då. Mer realistiskt är att kunna enas inom ett specifikt affärsområde av gemensamt intresse för parterna och därför sannolikt genomsyrat av relativt överensstämmande uppfattningar. Kanske väljer man att enas under en begränsad tidsperiod. Kanske väljer man att kontraktsskriva endast små informationsmodeller för avgränsade gemensamma syften efter annan indelning än affärsområde (t ex ämnes-område, ansvarsområde, ...). Samtliga informationsmodeller antas uttryckta med hjälp av samma överenskomna begreppsmodell (figur 23).



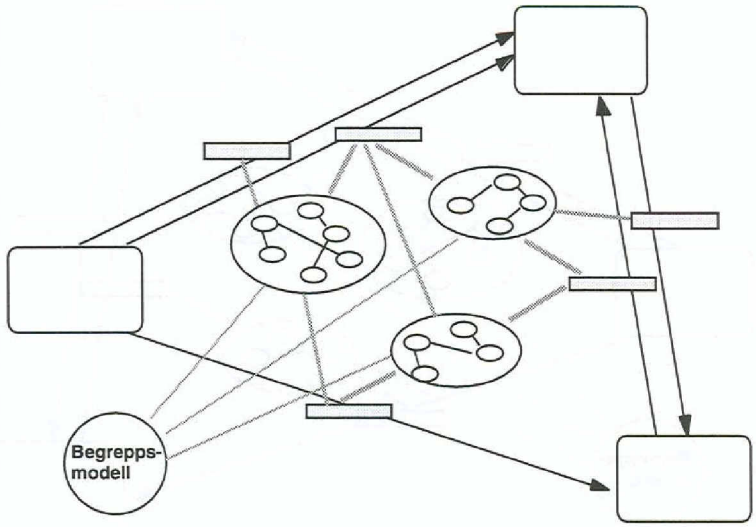
Figur 23

Med ett större antal parter inblandade och med en uppsättning olika informationsmodeller definierade, ökar sannolikheten att modellerna uttrycks med hjälp av olika begrepps-modeller. Detta kan i det enskilda perspektivet vara välmotiverat om de inblandade parterna för vissa behov är vana vid en viss begreppsmodell, om etablerade standarder finns för det avsedda ämnesområdet (t ex STEP/EXPRESS), om en viss begreppsmodell behövs för att uttrycka väsentliga semantiska aspekter, o s v. Se figur 24.



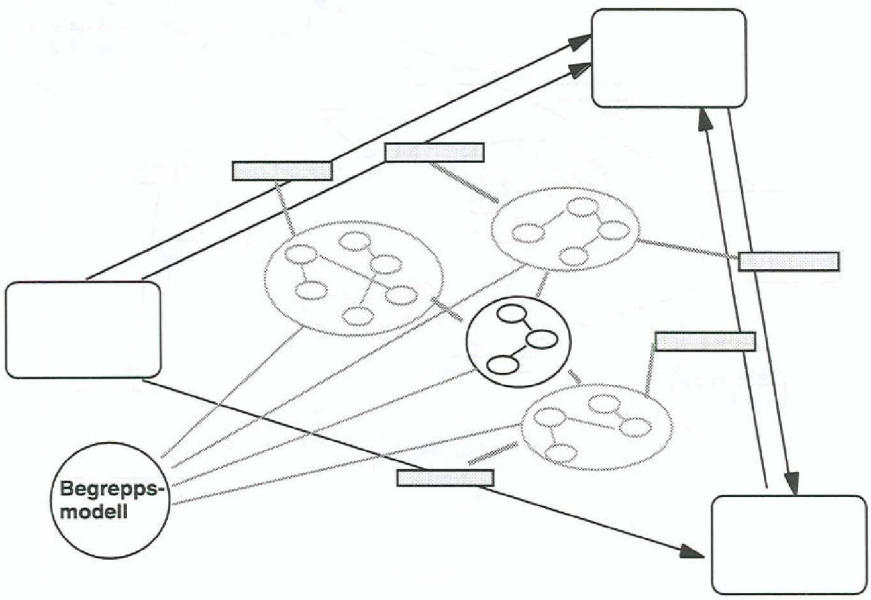
Figur 24

Givetvis får man betala med ökade svårigheter när det gäller informationsmodell-samordning, när så är önskvärt. Med ensad begreppsmodell skulle sådan samordning annars spontant kunna realiseras genom att för viss sändning, som berör flera informationsmodeller, referera till dem alla istället för att behöva etablera en ny samordnad informationsmodell (figur 25).



Figur 25

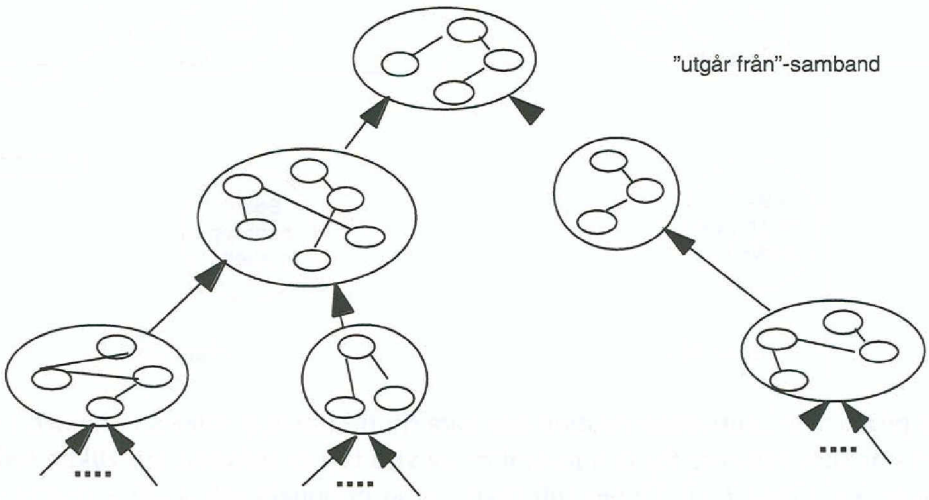
Om det visar sig att flera informationsmodeller i realiteten (fast kanske under olika benämningar) beskriver överlappande fenomen, är detta en indikation på möjliga samordningsfördelar. Kanske går det att steg för steg komma överens om en gemensam kärna som är okritisk för de skilda behoven, eller tvärtom, synnerligen avgörande för verksamhetens grundläggande idé. Finessen är att endast så mycket som man vill/orkar kompromissa om ligger i den gemensamma kärnan vid varje tidpunkt. Självfallet underlättas samspelet mellan den gemensamma modellen och de övriga om samma begreppsmodell används av dem alla. Figur 26 speglar den hypotesen.



Figur 26

Även mellanlägen mellan figur 25 och figur 26 är inom större utbytesmiljöer i högsta grad realistiska alternativ. Till dessa räknas överenskomna informationsmodeller mellan subgrupper av de samverkande. Dessa kan i sin tur baseras på en mer gemensamgjord kärna, som i sin tur .... . Olika grader av samordningshierarkier mellan informations-modellerna kan med andra ord bli resultatet. Återigen är användning av samma begrepps-

modell för dem alla en teoretisk önskedröm. Olika särintressen, roller, ansvar, styrning, behov m m för de olika informationsmodellerna innebär dock sannolikt i många fall att olika begreppsmodeller kommer till användning med åtföljande samordningsproblem.



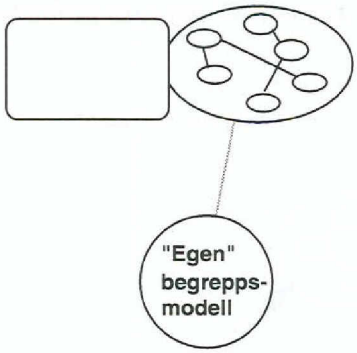
Figur 27

Observera att diskussionen ovan rört olika grad av generella överenskommelser kring informationsuppfattning. Oavsett vilken av ovanstående förutsättningar som gäller, måste därutöver respektive meddelandetyp (överföringstyp) preciseras i någon överenskommelse mellan sändare och mottagare, gärna enligt någon standardiserad mall.

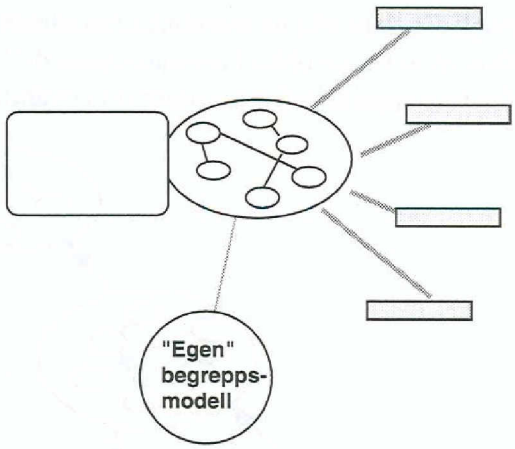
5.2 Enkla informationsutbyten, partstyrda överenskommelser

I många fall är informationsutbytet en del i en sammanflätad affärsprocess eller åtminstone av kritisk vikt för verksamhetens bedrivande. En mycket noga utformad överenskommelse eller kontrakt med alla upptänkliga förutsättningar och villkor inkluderade är ett måste. Informationsmodeller för meddelanden ligger i samtliga samverkande parters intresse och ansvar att ta fram med, så långt möjligt, hänsyn till egna behov. Alternativen i avsnitt 5.1 ovan speglar denna situation.

Därutöver kan det finnas utbyte av mera löslig karaktär där en informationstillhandahållare helt enkelt annonserar ut vilken information denne är beredd att tillhandahålla. Tillhandahållaren dikterar hur "annonsen" utformas och vilka övriga villkor som gäller, d v s hur informationsmodellen formuleras och hur gränssnittet för övrigt är utformat (figur 28 a). Informationskonsumenten preciserar sedan innehållet på det önskade informationsuttaget med hjälp av gränssnittsspråket. Något kontrakt att tala om existerar inte. Som en mer begränsad service tillhandahåller producenten endast en given uppsättning fördefinierade meddelandetyper baserade på och förklarade genom informationsmodellen (figur 28 b).



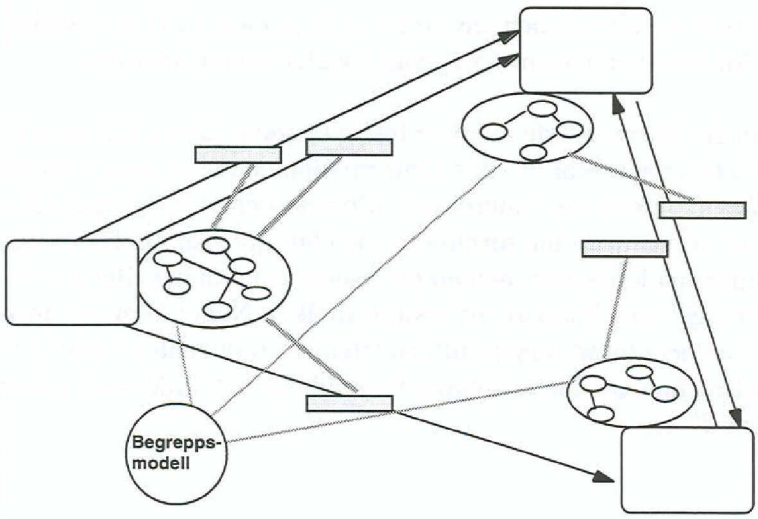
Figur 28 a



Figur 28 b

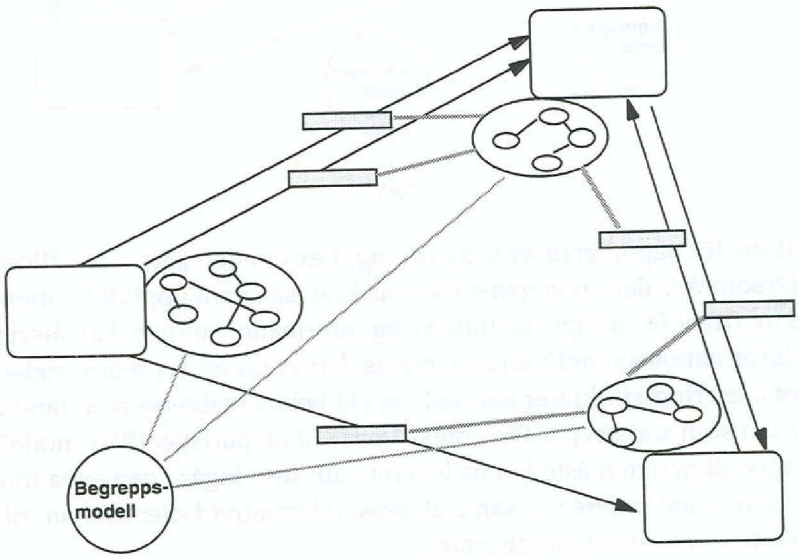
Denna producent-konsument-relation kan vara rimlig i många olika situationer, t ex där en part står som ansvarig för och insamlare av viss typ av information vilken sedan många önskar tillträde till. Öppen information i kommunala och statliga system svarar väl mot förutsättningarna. Kommunen ansvarar för informationen (många gånger genom att även vara ansvarig för den verksamhet som genererar informationen, t ex byggnadslov). Många kommuninvånare kan vara intresserade av en viss uppgift (t ex senaste kommunfullmäktigeprotokollet, en grafisk presentation av avloppsnätet, ...). Internet är urtypen på en miljö där denna förutsättning varit rådande i begränsad skala. Vanligt är standardiserade meddelandetyper, dokument, rapporter. Informationsmodellbaserat utbyte kan förväntas i allt större utsträckning framöver både mot tillämpningar och direkt mot databaser. Mer avancerade dialoger kan förväntas i gränssnittet.

Finns en etablerad begreppsmodell, som samtliga producenter "annonsörer" tillämpar, är det inga svårigheter för intresserade informationskonsumenter att rent faktamässigt tolka respektive informationsmodell. Är den mycket utförlig borde dessutom de flesta riskerna för semantiska feltolkningar vara raderade. Självfallet kan misstag och missuppfattningar uppstå. Detta är oundvikligt och åtgärdas successivt enligt "lära av misstagen"-strategin.



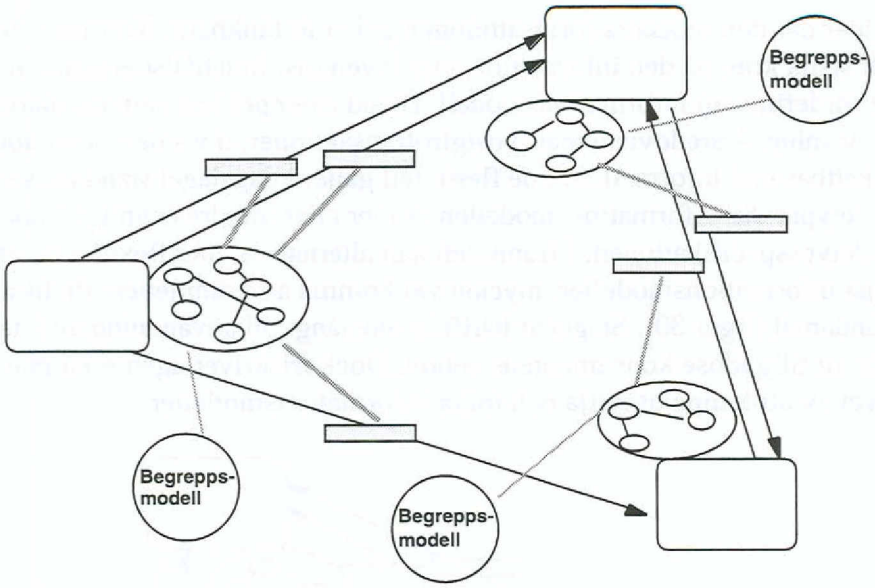
Figur 29

Även miljöer där den motsatta förutsättningen gäller är tänkbara. Konsumenten kan och har rätt att ställa krav på den information som levereras. Innehållsdeklaration ska här svara mot fördefinierad informationsmodell. Bland exempel kan nämnas datoriserad momsredovisning, årsredovisningar, postgirotransaktioner, d v s det mesta som tidigare varit blankettbaserad information. I de flesta fall gäller i dagsläget standardiserade meddelandetyper där informationsmodellen är mer eller mindre osynligt "inbakad" i meddelandetypsspecifikationen. I framtiden kan alternativa, mer flexibla överföringar, baserade på informationsmodeller, mycket väl komma att komplettera de fördefinierade för olika ändamål (figur 30). Steget är härifrån inte långt till användning av intelligenta agenter för att tillgodose konsumentens behov. Dock friskriver agenterna inte på något sätt behovet av att kunna utnyttja och tolka informationsmodeller.



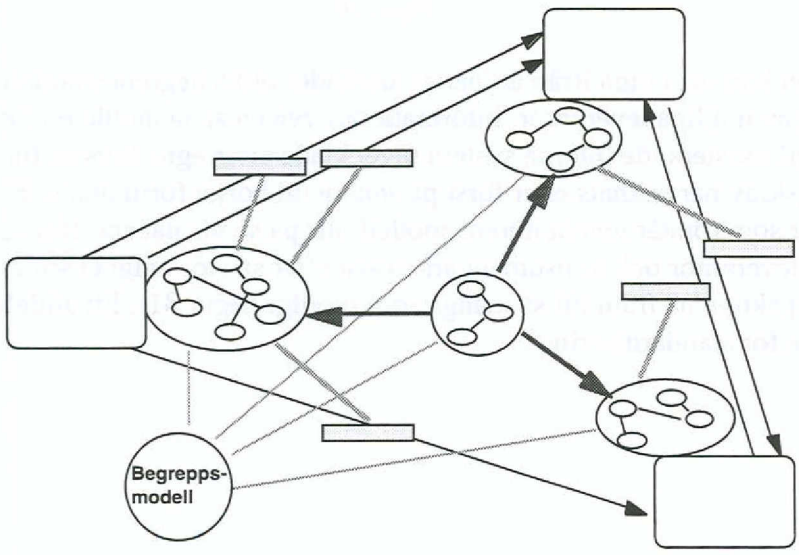
Figur 30

I normalfallet kan man utgå ifrån att parter använder olika begreppsmodeller för annonsering av möjliga leveranser. Informationsservicen är sannolikt en funktionell påbyggnad på existerande interna system utvecklade efter egna interna förutsättningar. Samordningskrav har saknats eller först på senare tid börjat formuleras. Endast de konsumenter som förstår leverantörens modell blir på så vis nåbara. Här ankommer det på både leverantör och konsument att sträva efter så stor enighet som möjligt för att nå ut till, respektive nå fram till så många som möjligt (figur 31). Ett alldeles påtagligt insatsområde för standardisering!



Figur 31

Där behov finns för någon grad av samordning, t ex om den partstyrda filosofin gäller inom ett affärsområde där vissa gemensamma krav ska vara uppfyllda, men där samtidigt en hög grad av frihet för övrigt ska råda kring informationsutbyte, kan återigen en gemensam informationsmodellkärna etableras. Utnyttjas en begreppsmodell som erbjuder specialiseringsstrukturer kan sådana vid behov etableras med lämpliga objekttyper ur kärnan som supertyper för objekttyper vid de partspecifika modellerna. En regel som säger att parten måste kunna leverera allt som ingår i den egna modellen plus de delar av kärnan som refereras, kan etableras. Alternativt fyller kärnan rollen enbart som en semantisk ensare och preciserare.



Figur 32

Inom en verksamhet borde det vara möjligt att komma överens om en gemensam begreppsmodell även mellan olika affärsområden. Under mer generella omständigheter bör anpassning till standarder eller väl spridda konventioner eftersträvas. Ämnesspecifika standarder för både begreppsmodeller, syntax och avgränsade ämnesområden

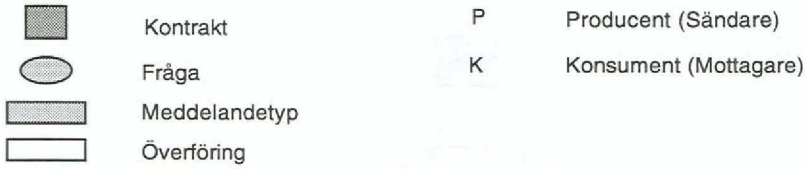
(subject areas) finns t ex när det gäller produktdata i STEP/EXPRESS-standarderna och när det gäller Case-data i CDIF-standarden.

Ett vedertaget neutralt gränssnitt är förstås SQL. Det baseras på relationsmodellen som begreppsmodell samt på en standardiserad syntax. Vidareutveckling pågår i form av SQL3 för att erbjuda både en rikare begreppsmodell och rikare uppsättning datatyper.

5.3 Andra dimensioner på informationsutbyte

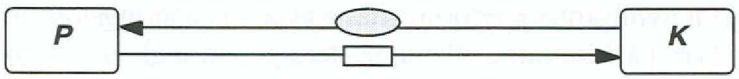
5.3.1 Engångsmeddelande – Abonnemang

Symboler använda i detta avsnitt:



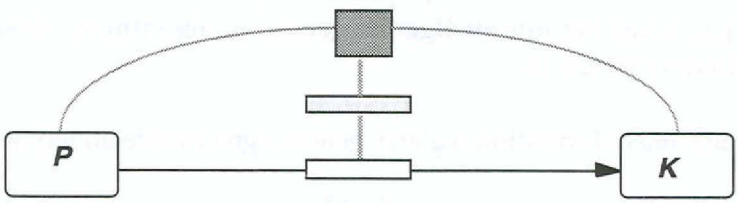
Antag att informationsmodell, m m är given mellan parterna.

Att ställa en fråga och få ett svar, t ex med hjälp av SQL och en överföringssyntax, är den enklaste formen av utbyte.



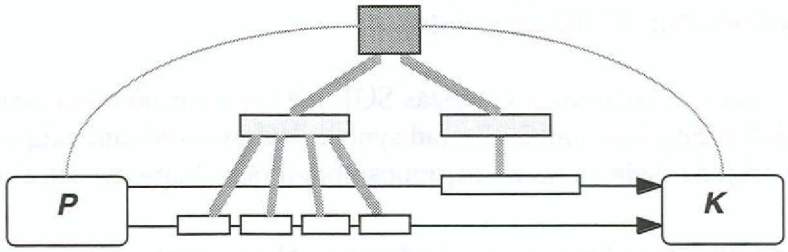
Figur 33

Även ett enstaka meddelande eller en meddelandesekvens (överföring) kan behöva kontraktskrivas. Det kan bero på att informationen är komplex till sin natur, berör icke-öppen information, är kvalitets- eller behörighetskänslig och av den anledningen behöver behandlas enligt speciella rutiner, ska utföras vid viss tidpunkt, kanske kostar pengar, ska krypteras, ...



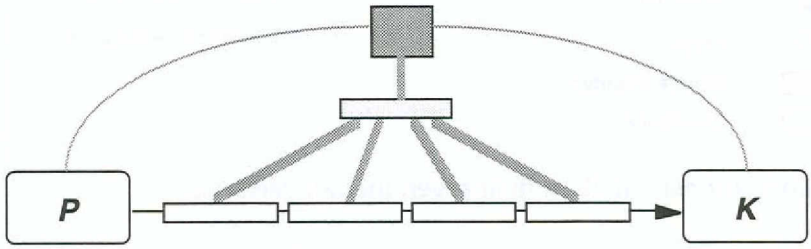
Figur 34

Upprepat informationsutbyte regleras regelmässigt genom en överenskommelse (kontrakt). Varianter att ta hänsyn till är om en meddelandesekvens alltid ska avse all relevant information eller endast nytillskott. Nytillskottsalternativet kräver förmodligen reglering av två meddelandetyper, dels den första totala överföringen och därefter tillskotten. Med tillskott följer behov av välfungerande versionshantering – inte minst hos leverantören, korrekt sekvensering av sändningar samt välintegrerad felhantering. Förutsättningarna är snarlika vanlig transaktionshantering i och med att en överföring rimligtvis bör resultera i ett konsistent informationsläge hos mottagaren.



Figur 35

Med total överföring undviks dessa problem (figur 36). Å andra sidan blir överföringen mer omfattande. Effekterna måste bedömas från fall till fall.



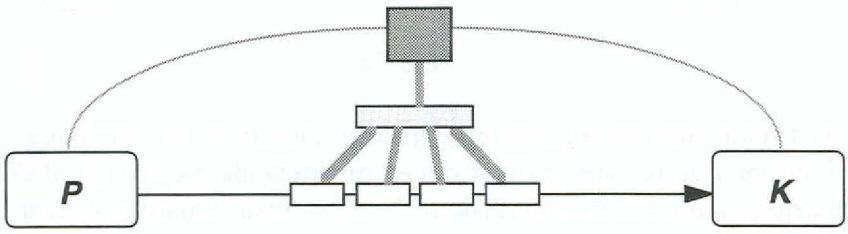
Figur 36

Vid både full- och nytillskottsöverföring måste även ren ändringsinformation hanteras. Ändringen kan bero på felrättning eller uppdatering (ändring). Har någon arbetat på och reviderat en konstruktion av något slag (ritning, CAD-spec, mjukvaruspecifikation, etc) kanske så mycket ändrats att i praktiken en full överföring blir mindre omfattande än ändringsinformationen, inkl diverse nödvändig styrinformation.

Återigen måste stor vikt läggas på korrekt sekvensering av överföringarna samt på att mottagaren har möjlighet att uppfatta om och när man befinner sig i ett konsistent informationsläge.

Observera att ändringar orsakade av nya/ändrade algoritmer på informationsmassan i framtiden lika gärna kan komma att åtgärdas genom att algoritmen skickas ut, som att de ändrade uppgifterna skickas ut.

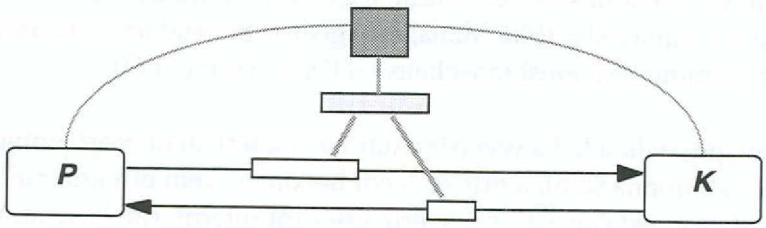
Överföring av ändringsinformation regleras genom egna meddelandetyper.



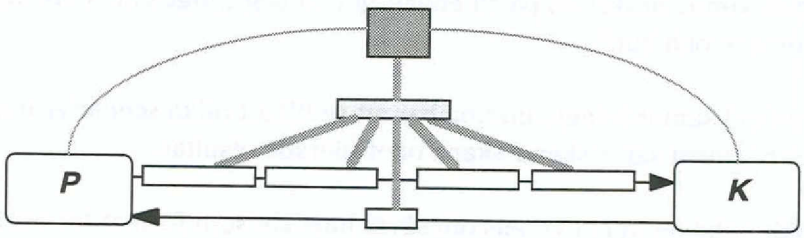
Figur 37

Kontraktsbaserade överföringar kan äga rum med viss periodicitet eller när vissa förutsättningar inträffat hos leverantören (händelsebaserat). De kan även initieras genom

en explicit begäran från mottagarens sida. En begäran kan vara enkel "Leverera enligt kontrakt xxx, meddelandetyp xxx". Begäran kan också vara dynamiskt villkorad, där villkoren och förutsättningarna skickas med begäran i en mer eller mindre omfattande överföring. Villkoren kan utgöras av en uppsättning ingångsparametrar, men även av en omfattande villkorsspecifikation. I det senare fallet måste ett överenskommet villkorsspråk existera. I figur 38 ger begäran upphov till en svarsleverans. I figur 39 innebär begäran snarare att leveranser fr o m nu ska baseras på en ny villkorsspecifikation, d v s meddelandeströmmen enligt kontraktet ska svara upp mot de nya utsökningsvillkoren. Kontraktet för övrigt är oförändrat.



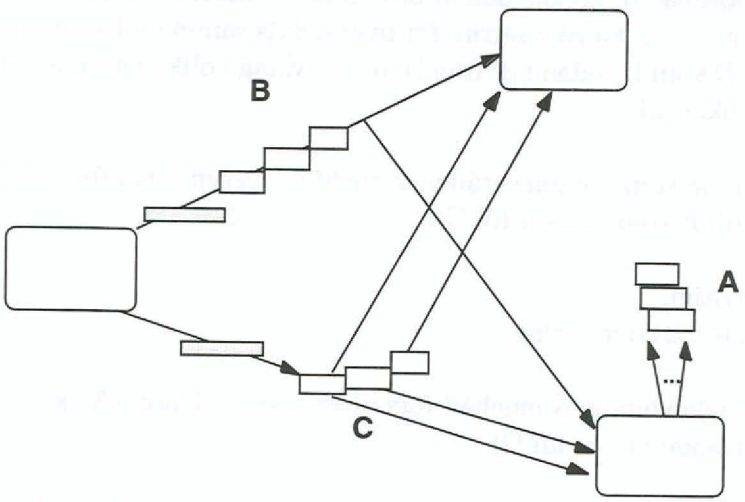
Figur 38



Figur 39

5.3.2 Broadcasting

En annan dimension på informationsutbyte uppstår när informationsutbytet tillåts ske enligt broadcasting-princip, d v s ett explicit utskick till alla som kan eller vill "lyssna" (A). Eventuellt kan "alla" vara begränsat till en viss uppsättning konsumenter per meddelandetyp (B). Alternativt kan begränsningen gälla specifika överföringar (C).



Figur 40

5.3.3 Kontrollerade, sammansatta informationsutbyten

Än mer svårhanterlig blir situationen om bearbetningssekvenser, arbetsflöden, scenarier (jmf Open EDI) ska kunna hanteras!

Informationsutbytet måste här regleras mellan flera parter där varje part har en definierad roll med nyanser baserade på aktuella tillstånd hos det samverkande "systemet". Bl a stipulerar en roll vilka typer av överföringar, från vilka, när, m m som ska accepteras samt vilken informationsbearbetning och vidaresändning som rollen ska effektuera. Ägare och ansvarig till informationselement måste regleras med hänsyn till var i transportkedjan elementet befinner sig. (Ska Manager/Agent-roller etableras på liknande sätt som tillämpas inom kommunikationsbranschens GDMO-standarder?)

Ett scenario kan vidare innehålla precision om vilka alternativa bearbetningsvägar som står till buds för att uppnå samma effekt. Vem bedömer, vem effektuerar? Kanske ska olika typer av villkor testas per steg (externt eller rollinternt, tidsbaserat, beroende på inträffade händelser, explicita begäranden från någon part, ...).

Scenarier måste kunna beskrivas på ett enhetligt och begripligt sätt. Vem ansvarar för dess genomförande och hur?

Någon överordnad samordningsfunktion har att se till att olika scenarier inte strider mot varandra eller på annat sätt riskerar skapa odefinierade resultat.

Alla inblandade behöver vara överens om såväl innehåll som format för överföring. Samma begreppsmodell används rimligtvis av samtliga parter inom ett scenario.

Att etablera och underhålla välfungerande, sammansatta informationsutbyten är med andra ord synnerligen komplicerade aktiviteter. Standarder saknas, likaså erfarenheter av mer ambitiösa ansatser.

5.3.4 Okontrollerade sammansatta informationsutbyten

Ett scenario beskriver en ordnad mängd bearbetningssteg. Styrning, kontroll, uppföljning är möjlig. I en oordnad miljö kan den information A sänder till B föras vidare till C och D som i sin tur kan ..... . C kanske senare för över delvis samma information, eventuellt förvanskad, till D som D redan fått från B, o s v. Många olika varianter av vidaresändning kan tänkas, bl a

a. All information som kommer från viss meddelandetyp förs oförvanskad vidare. (A skickar till B som skickar till C).

   a1. Regelmässigt.
   a2. Avseende viss överföring.

b. Del av meddelandetypens innehåll förs oförvanskat vidare. (A skickar till B som skickar delmängd vidare till D).

c. Meddelandetypen oförvanskad kompletteras med annan information. (A skickar till B som skickar till C som lägger till något och skickar till D).

d. En kombination av b och c på så vis att vissa delar av en meddelandetyp förs samman med annan information i en ny meddelandetyp.

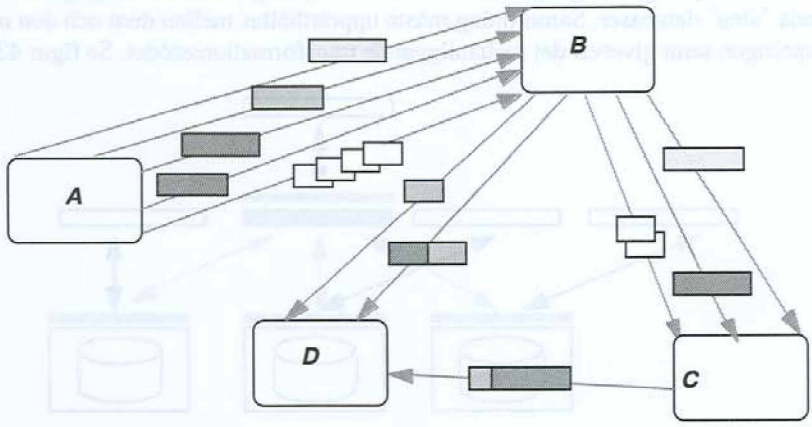
e. Endast vissa förekomster enligt meddelandetypen förs vidare, t ex endast de order som har en slutsumma över 10.000.-. (A skickar fyra förekomster enligt en meddelandetyp till B som skickar två av dem vidare till C).

f. Variant av d. Den mest "lösaktiga" kopplingen uppstår när meddelandeinnehåll "löses upp" enligt mottagarens godtycke för att sedan helt eller delvis sammanställas i någon "ny" meddelandetyp. Alternativet erbjuder ingen möjlighet till spårning av informationen.

Alternativ d är sannolikt mycket vanligt.

Hur upptäcka att det är samma information eller att vissa uppgifter erhållits på basis av kända uppgifter, alternativt inte baserats på det underlag man förmodat? Hur upptäcka att en uppgift är en förvanskning av en annan? Hur stor är risken för att duplikat "slinker igenom", speciellt om inte en enhetlig informationsmodell används? Hur vet man att en utelämnad uppgift innebär att den inte finns och inte beror på något fel eller på nonchalans? Hur, var, vem kontrollerar att informationshanteringen i praktiken utförs i enlighet med intentionerna? Hur garanteras korrekt koordination? Bör varje uppgift "stämplas" med utgivare och tid?

Hur förfara med uppdateringar? Vem är ansvarig för en modifierad information, den som modifierat eller den som ansvarar för ingångsvärdena? Båda? Om ingångsvärdena visar sig vara felaktiga och modifieraren handlat i "god tro"? Med mycket mera att fundera över.

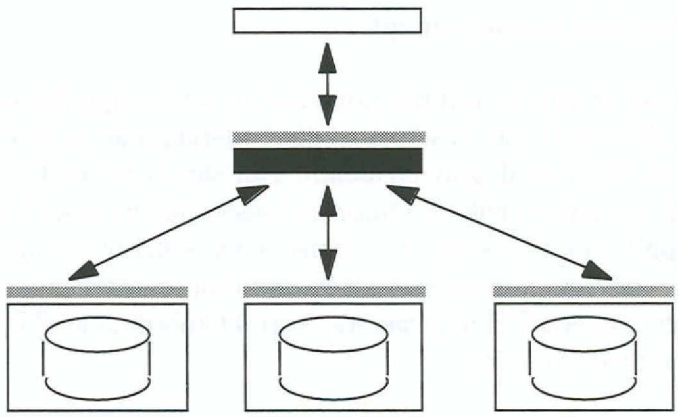


Figur 41

Okontrollerat informationsutbyte kan mycket väl vara acceptabelt under vissa förutsättningar. Man bör dock känna till förutsättningar, varianter och risker.

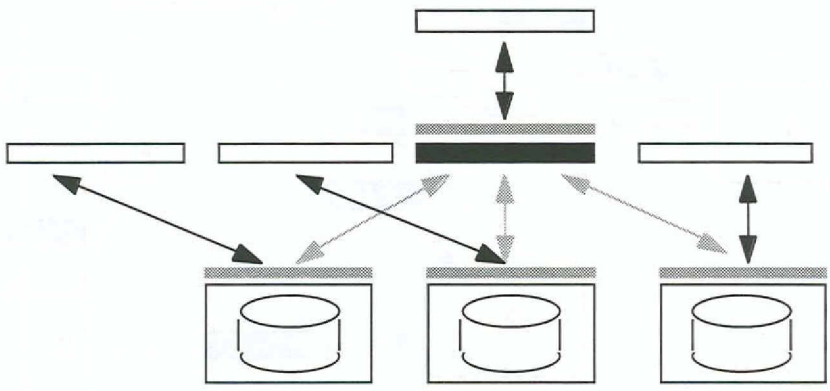
6. Samordning av gränssnitt för olika dbms-miljöer

Olika databasmiljöer erbjuder olika gränssnitt och begreppmodeller. Tillämpningars behov av att operera på data i flera fristående databaser kan förväntas öka markant. En naturlig strävan bör naturligtvis vara att försöka etablera en enhetlig informationsmodell mot tillämpningen och överlåta till någon servicemekanism att sköta den interna samordningen mellan databaserna. Mekanismen måste klara olika dbms-gränssnitt, de olika databasernas informationsmodeller och eventuella överlappningar och/eller inkonsistenser i databaserna. Sådana verktyg finns.



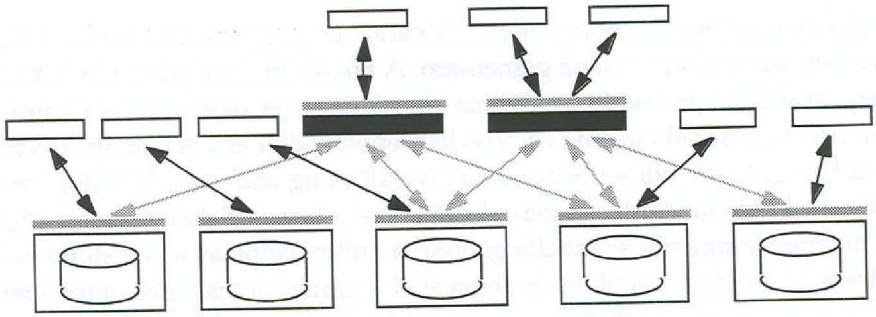
Figur 42

En komplicerande faktor är förstås att de ursprungliga databaserna skapats för sina specifika tillämpningar en gång i tiden. Dessa tillämpningar önskar antagligen fortsätta använda "sina" databaser. Samordning måste upprätthållas mellan dem och den nya tillämpningen samt givetvis det mellanliggande transformationsstödet. Se figur 43.



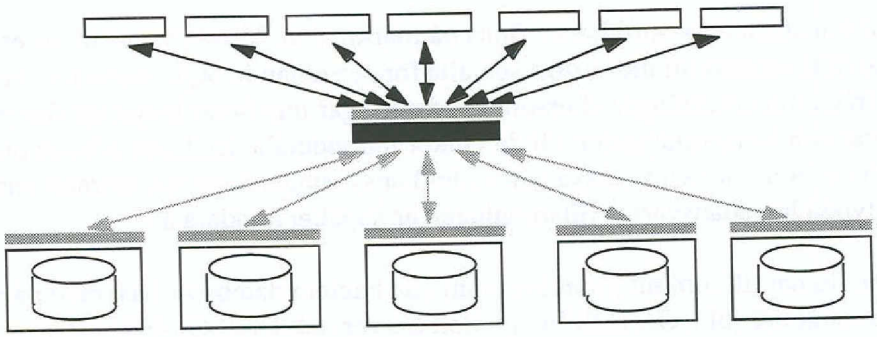
Figur 43

Situationen blir än mer besvärlig om vi betänker att fler än en tillämpning arbetar mot en kombination databaser via transformationsstöd (förhoppningsvis samma, men inte säkert), dessutom kanske fler än en tillämpning per databaskombination (informations-modell).



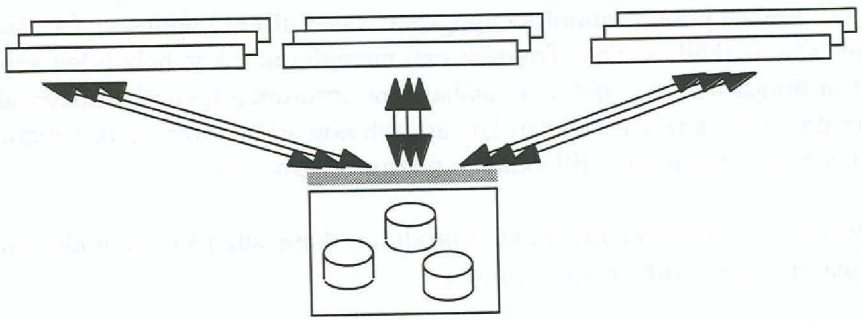
Figur 44

Ett övergripande enhetligt gränssnitt och vidhängande informationsmodell vore återigen självfallet ett mål att sträva efter (figur 45).



Figur 45

I realiteten är vi då återigen mycket nära en lösning med en enda logisk databas enligt figur 46 under förutsättning att redan existerande databaser går att transformera till (se avsnitt 7) och inordna i den resulterande databasen och dess informationsmodell. Där finns ett antal tillämpningar, var och en i ett antal klient-kopior. Databasen är under kontroll av en databashanterare som medger distribution, replikering av data samt parallell och delegerad exekvering.



Figur 46

Dock är detta en teoretisk idealbild som knappast är realiserbar i större verksamheter. Olika systemgenerationer, samordningsproblem, behov av snabba anpassningar till nya verksamhetskrav, ansvarsfrågor, m m är hindrande realiteter. Vad göra?

Ju färre alternativa typer av dbms med tillhörande begreppsmodell desto enklare att samordna och göra informationen gemensam. Å andra sidan har vi också kunnat konstatera att olika typer av dbms alla har sina förtjänster inom vissa tillämpningsområden. En onyanserad ensning till viss begreppsmodell är i det perspektivet alltså inte heller alltid av godo. Samtidigt bör onödig diversifiering undvikas. I möjligaste mån bör miljöer baserade på standarder skapas. Används relationsmodellen bör t ex tillgängliga databashanterare kunna tillhandahålla gränssnitt enligt någon av de existerande SQL-standarderna. Samtidigt kan givetvis andra skäl kopplat till viss leverantör spela in (leverantörsrelation, serviceförmåga, ekonomisk stabilitet, pris, ...).

I vad mån äldre databaser bör migreras till lämpligare begreppsmodeller och dbms eller endast beläggas med ett tilläggsgränssnitt måste bedömas från fall till fall. Migrering diskuteras vidare i avsnitt 7.

Ett antal transformationsprodukter finns på marknaden. Vissa syftar till att erbjuda ett enhetligt SQL-gränssnitt mot i stort sett alla förekommande SQL-varianter. Främst är det här fråga om utsökningar eftersom uppdateringar innefattar betydligt fler problemställningar och hänsynstaganden till de villkor individuella databaser arbetar under. Dessutom tillkommer semantiska och andra hänsynstaganden som normalt endast den för datatypen huvudansvariga tillämpningen är kapabel att klara av.

På senare tid har tillkommit produkter som kan hantera databaser baserade på olika begreppsmodeller (bl a ONTOS Integration Server och UNISQL/MM). Det erbjudna gränssnittet baseras normalt på någon variant av semantisk datamodell (ER-modell). Ofta används begreppet objektmodell för att antyda en avancerad miljö enligt "senaste skriket". Inte minst det ökande intresset för data warehouse-miljöer kommer att generera fler och alltmer avancerade produkter.

Det är knappast tillrådligt att försöka konstruera en egen transformationsmekanism. Mekanismen är i sig mycket komplicerad med ansenliga felrisker. Felaktigheter kan ge allvarliga konsekvenser genom att vitala data kan bli felaktiga eller missvisande. Ett extra bekymmer är att sådana fel många gånger kan hålla sig "dolda" för upptäckt under ansenlig tid. Endast gravt osannolika uppgifter "lyser direkt i ögonen". Ett lika stort bekymmer är underhållsbördan. Transformationsreglerna måste hela tiden anpassas till nya förutsättningar hos käll- och måldatabaserna, förutom allt vanligt underhållsarbete som följer på ett programvaruansvar. Det är jobb nog att underhålla informationsmodellen och dess anpassning till aktuella tillämpningsbehov.

Med andra ord, rekommenderas standardprodukter baserade på etablerade standarder eller sådana med god marknadspenetration.

7. Migrering från dbms typ x till dbms typ y

7.1 Varför, när?

Avsnitt 6 diskuterade anpassning av en tillämpning mot ett antal databaser av olika typ via ett gemensamt gränssnitt tillhandahållet av lämpligt anpassningsverktyg. Dessa verktyg erbjuder normalt mycket avancerad funktionalitet. Dock måste med nödvändighet vissa ovan noterade begränsningar vad gäller uppdateringar accepteras. Av den och andra anledningar kan en anpassning av en tillämpning till ny databasmiljö vara ett intressant alternativ.

Byte mellan olika SQL-varianter är normalt genomförbara på ett kontrollerat sätt eftersom i grunden samma begreppsmodell utnyttjas. Dock visar erfarenheter att anpassning även till olika varianter på "samma tema" inte sällan ställer till överraskande många problem. Dels kan de semantiska uttrycksmöjligheterna skilja, dels erbjuder de olika miljöerna normalt olika typer av funktionalitet och tekniska lösningar som måste kunna överbryggas.

Säkerligen existerar idag även många tillämpningar baserade på egenutvecklad, hierarkisk eller relationsmodell, som istället "skulle må bra av" att baseras på objekt- eller ordbms-modell. En orsak kan vara att tillämpningen har egenskaper som svarar väl mot den service nya dbms-ansatser tillhandahåller men som inte fanns att tillgå när systemet utvecklades. I andra fall kan ett behov av en modernisering (multimediabasering, komplexare datastruktur, utökad funktionalitet, etc) av en tillämpning vara den tillräckliga katalysatorn för en omskrivning i ny dbmsmiljö. Behov av ensning mot etablerad systemutvecklingsmetod eller uttalad policy inom företaget kan vara andra skäl. Kanske gäller en OO-baserad metod med tillhörande OO-Case-stöd. Kanske gäller policyn att C++ eller Smalltalk ska användas genomgående. En integration mellan språkmodell och dbms-modell underlättar samverkan dem emellan. Kanske gäller viss trend inom den egna branschen. Kanske har man i en allmän moderniseringsiver fastslagit att all nyutveckling ska vara objektorienterad eller hybridorienterad i största allmänhet.

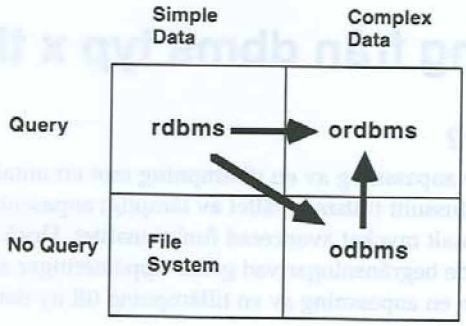
Gäller endast det sista argumentet kan dock moderniseringen visa sig kosta betydligt mer än den smakar. En mycket noggrann kalkyl baserad på ett helhetsperspektiv och i alla delar realistiska antaganden utgör ett synnerligen centralt beslutsunderlag.

Utgår vi från Stonebrakers uppdelning i dbms-typer och bortser från "rutan" File System (eftersom begreppsmodell och andra förutsättningar där är okända) framstår tre migreringsalternativ som mest sannolika:

a. att gå från rdbms- till ordbms-basering.

b. att gå från rdbms- till odbms-basering.

c. att gå från odbms- till ordbms-basering.



Figur 47

7.2 Mellan vilka, hur?

Oavsett alternativ gäller att den gamla och den nya begreppsmodellen har varierande semantisk uttryckskraft, som på något sätt måste kunna hanteras vid transformeringen av informationsmodellerna. Nedan följer några kommentarer kring var och en av de möjliga alternativen (inklusive de mindre sannolika mot pilarnas riktning).

a. Rdbms -> odbms

Med referens till de diskuterade trenderna i avsnitt 2 finns sannolikt mycket sällan tillräckliga skäl att gå från en rdbms- till en odbms-miljö. Odbms är inte "nästa generation dbms", utan en miljö som profilerat mot att svara upp mot vissa specifika behov – funktionalitet som rdbms inte erbjuder, i alla händelser inte lika elegant. Transformation av en "ordinär" rdbms-tillämpning kan möjligtvis vara motiverad av stränga samordningsskäl eller om tillämpningen ska ändras/vidareutvecklas i en riktning som odbms men inte rdbms smidigt klarar av att realisera. (Vi bortser här från tillämpningar som alldeles naturligt platsar i odbms-kategorin men av någon orsak ändå blivit realiserade i rdbms-miljö.) Transformationen kräver en övergång till ett helt nytt modelltänkande, något som säkerligen kräver både ansenlig tid, motivation och kunskap. Byte av programmeringsspråk behövs i de flesta fall. Med andra ord, är det fråga om en mer eller mindre total omskrivning av tillämpningen.

b. Odbms -> rdbms

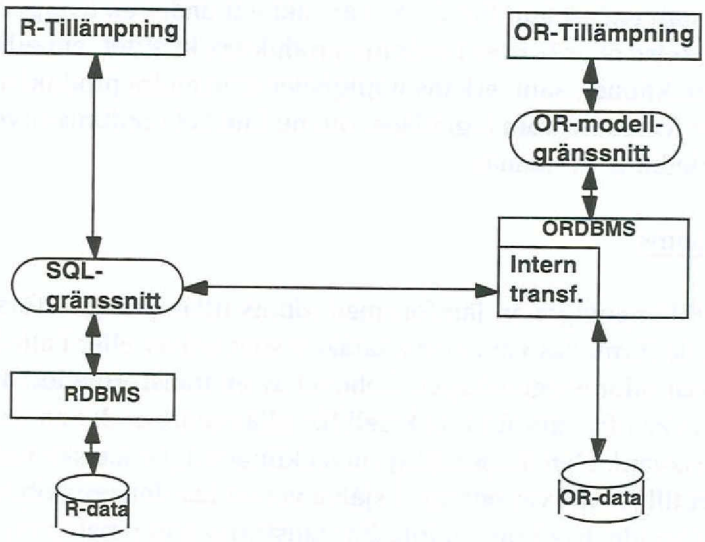
Kan mycket sällan bedömas motiverat. Visserligen finns en trend mot att odbms-tillämpningar alltmer kompletteras med "normal" databashantering, eller att databaserna får en sådan varierad användning att en odbms-lösning blir opraktisk eller, i vissa fall, onödig. I det läget ligger snarare en ordbms-lösning än en rdbms-lösning till hands.

c. Rdbms -> ordbms

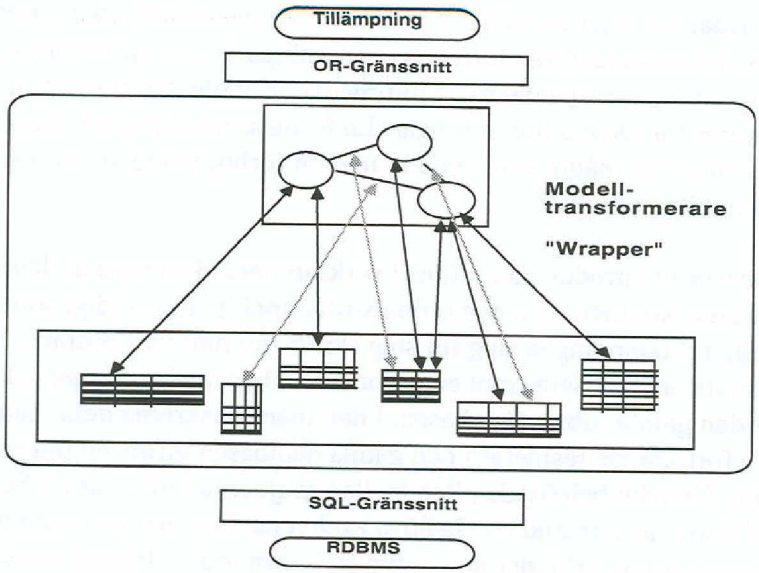
De ordbms-modeller som i dagsläget har störst genomslag är i princip fullt kompatibla med SQL. Att köra en rdbms-tillämpning med hjälp av en ordbms borde därför inte erbjuda alltför stora problem. Å andra sidan har man då heller inte vunnit något. Vinsten uppstår först genom en ommodellering där ordbms-modellens semantiska kraft och större variation av datatyper tagits i anspråk. Den gamla och den nya modellen kommer i de flesta fall att ha stora olikheter. Detta tillsammans med en i övrigt avsevärt utökad

funktionsrepertoar kommer normalt att resultera i en resurskrävande omprogrammering. Många rdbms-tillämpningar kan förmodas genomlöpa denna uppdatering. Inte minst nya krav på funktionalitet, prestanda, nya multimediabaserade datatyper m m kommer att vara katalysatorer för sådana investeringar. En förutsättning är dock framväxten av en stabil ordbms-standard (något som ännu är mer en förhoppning än en realitet) samt tillgången på stabila produkter.

En väg att successivt introducera ordbms i verksamheten kan vara att låta existerande tillämpningar leva parallellt med nya ordbms-tillämpningar samtidigt som valda delar av den ursprungliga tillämpningen steg för steg skrivs om mot nytt ordbms. Se figur 48. De delar som skrivits om realiseras mot en ordbms-databas medan övriga delar på vanligt sätt körs mot den gamla rdbms-databasen. I den mån omskrivna delar behöver kontakt med data som fortfarande residerar i den gamla databasen utförs en intern, automatisk transformation. Den dubbelriktade pilen mellan miljöerna indikerar att SQL-operationer i den gamla tillämpningen ibland kan behöva exekveras mot den nya ordbms-databasen. Vissa ordbms-produkter erbjuder denna typ av intern, dubbelriktad transformation. I teorin en enkel lösning, i realiteten en avancerad och riskfylld men dock realiserbar strategi. Inte minst transformationen erbjuder delikata problem, speciellt över ett längre perspektiv med anpassning till olika icke koordinerade versioner av scheman för såväl den gamla som den nya databasen. Se figur 49.



Figur 48



Figur 49

d. Ordbms -> rdbms

Detta är en väg som endast kan förväntas vara aktuell under en övergångstid som ett resultat av besvikelse över brister i ordbms-produkters kvalitet, enhetlighet, databas-administrativa funktioner, samverkansmöjligheter med andra produkter, m m. På sikt kan dessa bekymmer förmodas vara åtgärdade, om nu inte hela ordbms-utvecklingen på något sätt av någon anledning avstannar.

e. Odbms -> ordbms

Antalet odbms-tillämpningar är, jämfört med rdbms-tillämpningar, försvinnande litet. I de flesta fall får de förmodas vara av en karaktär som kräver eller i alla händelser mycket fördelaktigt passar odbms-egenskaper. Behovet av en transformation är med andra ord obetydligt. Sådan kan framförallt bli aktuell för tillämpningar där en mer semantisk modell anses önskvärd. Den ursprungliga utvecklingen ägde kanske rum vid en tid där det enda alternativet till rdbms var odbms. I själva verket har den rena objektmodellens egenskaper inte varit en idealisk kompanjon. En transformation innebär ett paradigmskifte som sannolikt kräver avsevärda insatser i ommodellering, omprogrammering och kompetensutveckling. Denna resursinsats kan dock vara motiverad, inte minst om samverkande tillämpningar baserats på ordbms-lösningar. Observera att en försvårande faktor återigen är dagslägets brist på standardiserad, enhetlig ordbms-standard.

f. Ordbms -> odbms

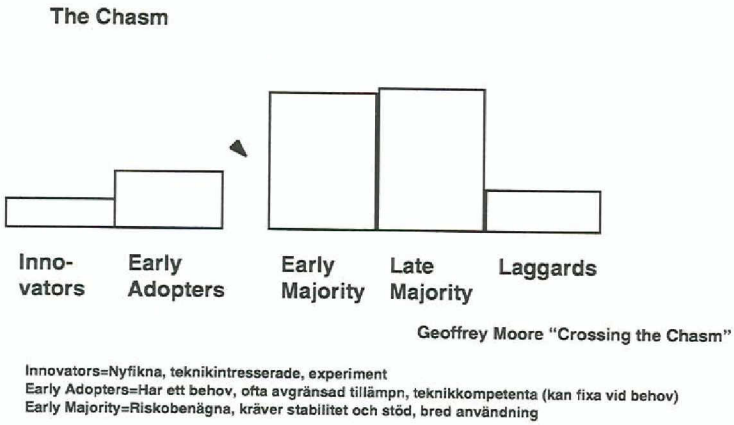
I stort sett samma argument gäller här som för alternativ a, med tillägget att behovet av transformation är om möjligt än mindre eftersom ordbms-modellen är betydligt mer semantiskt rik än rdbms-modellen. Även relativt odbms-orienterade tillämpningar kan hanteras med elegans.

Helt ny objektmodell

Mycket pekar på att ordbms-modellen kommer att få starkt genomslag till förfång för de övriga modellalternativen. Därmed inte sagt att ordbms-modellen är något idealiskt redskap, snarare en rimlig evolution utifrån en rdbms-horisont, dessutom i mittfåran av de kommersiella krafterna. Det är knappast osannolikt att någon mer renodlat semantisk datamodell så småningom kommer att se dagens ljus som ett alternativ till ordbms-modellen. Inte heller osannolikt kommer en mer renodlad objektmodell att etablera sig på databashimlen. Java m fl ansatser banar här vägen.

8. Avrundning

I dagsläget finns knappast några nästa generations databashanterare. Vad som finns är olika typer av dbms, var och en med sina kvaliteter och brister. De passar bra för vissa tillämpningar, mindre bra för andra och kan vara katastrof för ytterligare andra. Kontentan blir att en tänkt migrering till en ny dbms-filosofi måste bygga på en gedigen kunskap om de olika alternativens styrka och svagheter – måste föregås av en mycket seriös analys avseende effekterna för just den tilltänkta tillämpningen. Därutöver måste givetvis en generell produkt- och leverantörsbedömning göras kring kvalitet, funktionsrepertoar, stabilitet, standardanpassningar, långsiktig överlevnad, o s v. Att hålla i minnet är att såväl odbms som ordbms ännu befinner sig i de första faserna av sin utvecklingscykel. Applicerat på Geoffrey Moores indelning i "Crossing the chasm" har de ännu inte passerat den viktiga och svåra klyftan mellan "Early Adopters" och "Early Majority", med de konsekvenser för riskkalkylen detta må ha.



Figur 50

Observera dock att de flesta etablerade rdbms-leverantörer sannolikt kommer att utveckla sina produkter i ordbms-riktning framöver. Någon variant av ordbms-lösning kommer därför på sikt att bli norm. I dagsläget saknas dock stabilitet kring standard och produkter. Den presumtive kunden har därför att bedöma om man ska ta steget nu till en mindre ordbms-leverantör eller vänta till ett stabilare läge uppnåtts.

Vid samverkan genom informationsutbyten krävs inte samma typ av marknadsbedömning och överväganden. Här ligger problematiken kring en förståelse av problemsituationens alla komponenter och de olika varianter som står till buds samt att det i lika hög grad är ett semantiskt som ett tekniskt problem. Så långt möjligt bör standardprogramvaror användas.

Vad kommer då att hända med alla så kallade "legacy systems"? Grundtipset är att de kommer att lämnas orörda men att ökad öppenhet mot omvärlden kommer att realiseras genom enhetliga makrogränssnitt, genom olika former av "object wrapping" och genom ökade möjligheter till informationsutbyte med omvärlden. Att skriva om dem för anpassning till ny databasteknologi är i de flesta fall så resurskrävande och riskfyllt att det är försvarbart endast i sällsynta fall. Däremot blir läget givetvis ett helt annat om och när systemet ändå står för en total översyn och modernisering.