

# **Object Management Group (OMG)**

**– en översikt**

*Stig Berild*

©TRIAD December 1993



# Innehåll

## Förord 2

## 1. Upprinnelse, syfte 3

## 2. OMGs uppbyggnad 5

2.1 Medlemskap 5

2.2 Organisation 5

2.3 Synpunkter 7

## 3. Standardiseringsarbete 8

3.1 Vad är av intresse för standardisering? 8

3.2 Standardiseringsprocessen i stort 10

3.3 Synpunkter 10

## 4. Framtagna "standarder" 12

4.1 Object Model 12

4.2 (CORBA 1.1) 15

## 5. Pågående arbete 19

5.1 Arbete inom OSTF 19

5.2 Arbete inom ORBTF 20

5.3 Common Facilities 21

5.4 Löpande 21

## 6. Milstolpar 23

## 7. Avslutande synpunkter 24

# Förord

Syftet med denna rapport är att ge en översiktlig, allmän information om en, inom objektorientering, mycket uppmärksammas sammanslutning och de aktiviteter den bedriver. För den som vill "gå på djupet" finns flera skrifter utgivna av OMG samt givetvis möjligheten att bli medlem. Mera om detta nedan. Bakgrundsmaterial är OMGs egna publikationer.

Obs! OMG är mycket livskraftigt, vilket gynnar framväxt och spridning av nya eller reviderade resultat. Samtidigt begränsar detta givetvis ganska snabbt informationsvärdet i denna rapport.



# 1. Upprinnelse, syfte

Object Management Group (OMG) etablerades i april 1989 av åtta företag: 3Com, American Airlines, Cannon, Data General, Hewlett-Packard, Philips Telecommunications N.V., Sun Microsystems och Unisys med syftet att stödja och stimulera utveckling av och intresse för objektorienterad mjukvaruutveckling. OMG är en icke vinstdrivande organisation med huvudsäte i Framingham Massachusetts. Inom det både brokiga och vildvuxna objektorienteringsområdet visade sig snabbt OMG fylla ett stort behov. Organisationen har nu ca 340 medlemmar. Antalet stiger.

OMG noterar att framtiden innebär höggradigt distribuerade datorresurser av hög kapacitet. Problemen återfinns på mjukvarusidan med alltför resurskrävande utveckling, ineffektivt, riskfyllt underhåll och mycket begränsade möjligheter till flexibel anpassning till föränderliga behov. Information blir en alltmer kritisk resurs för överlevnad och framgång. Information måste kunna delges, bearbetas och förädlas där behoven finns. Information kommer att behöva hanteras på många olika sätt, av olika tillämpningar i den egna organisationen (egenutvecklade, köpta, tillfälligt nyttjade) eller av externa tillämpningar inom organisationer med vilka man samarbetar (underleverantörer, kunder, myndigheter, ...).

En förutsättning för denna vision är en gemensam semantisk förståelse för informationen hos dem som kommer i kontakt med den.

En annan förutsättning är att tillämpningarna måste kunna samverka med minsta möjliga bekymmer även om

- deras implementering ändras eller kompletteras
  - sättet att samarbeta med andra tillämpningar förändras
  - de flyttas mellan datormiljöer
  - de kommer och går, d v s byts ut
  - de återanvänds för nya/fler behov
- o s v

För effektiv samverkan behövs standardiserade samarbetsprinciper mellan tillämpningarna.

OMG berör för närvarande inte den semantiska aspekten. Inriktning är mot tillämpningssamverkan, d v s det man kallar interoperabilitet i en distribuerad och heterogen miljö. Som en grundidé ligger uppfattningen att denna interoperabilitet bäst kan uppnås genom ett objektorienterat synsätt, där tillämpningarna uppträder och agerar som objekt.

För att visionen ska bli realitet måste samverkan ske enligt en allmänt accepterad filosofi. Filosofin formuleras inom OMG i form av utgivna specifikationer. Den

allmänna acceptansen uppnås om specifikationerna anammas av tillräckligt många, tillräckligt inflytelserika på marknaden. En naturlig strävan är därför att kontinuerligt bredda medlemskretsen samt att utforma specifikationsarbetet på ett sätt som medger aktivt inflytande från medlemmarnas sida. Om både arbete och resultat inger förtroende, även utifrån ett externt perspektiv, etableras efterhand så kallade de facto standarder.

## 2. OMGs uppbyggnad

Vi har ansett det vara av vikt att ganska utförligt beskriva OMGs organisation och arbetsprinciper. På så vis kan var och en i framtiden bättre bedöma framtagna standarder, deras bakgrund, förutsättningar, marknadsrelevans m m. Samtidigt introduceras några begrepp och förkortningar. Dessa förekommer rikligt i OMG-dokument och artiklar om OMG.

### 2.1 Medlemskap

Det kan vara intressant att betrakta rollspelet och maktfördelningen bland OMGs medlemmar.

Medlemmarna (företag och organisationer) indelas i fem kategorier:

- a. Corporate member, som per år betalar mellan \$5.000-50.000 beroende på omsättning.
- b. Associate member, som per år betalar mellan \$1.000-15.000 beroende på omsättning.
- c. End-User member, som per år betalar \$10.000.
- d. University member, som per år betalar \$500.
- e. Subscribing member, som per år betalar \$1.000.

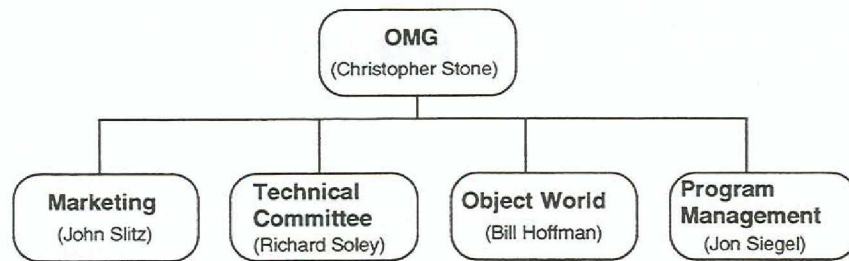
Svenska medlemsföretag per juni 1993 är (med ursäkt för eventuella fel):

- ABB Automation AB
- Ellemtel Utvecklings AB
- Enator Functional Systems
- Ericsson Programatic Sweden AB
- Mimer Software AB
- Objective Systems
- Sveriges Verkstadsindustrier
- Telia Data AB
- Volvo Data AB

### 2.2 Organisation

Den övergripande makten inom OMG ligger hos en styrelse (BOD=Board of Directors) med representanter från bland annat flera av de större medlemsföretagen.

OMGs vardagliga verksamhet utförs av en mindre grupp anställda, indelade enligt figur 1. Aktuellt ansvariga personer har inkluderats. Det kan vara bra att veta deras roller eftersom de ofta figurerar i olika konferenssammanhang och i samband med citat m m.



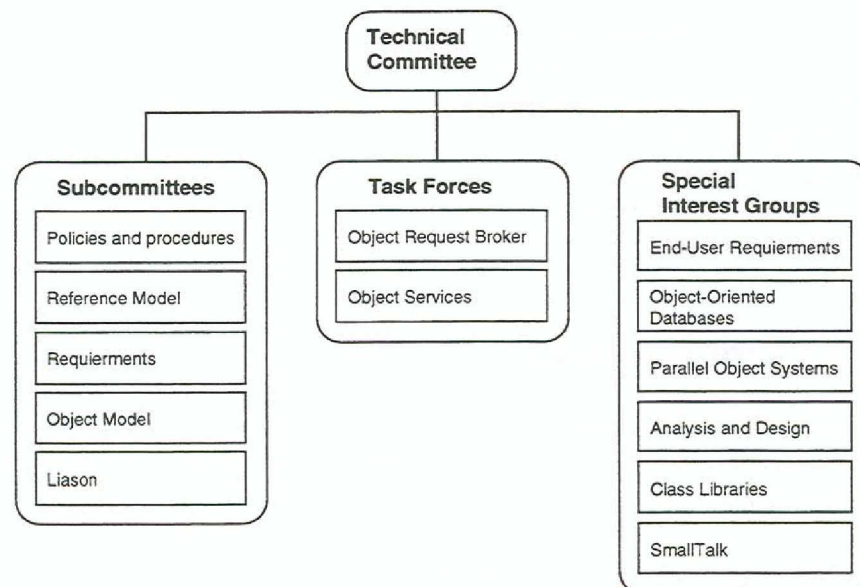
Figur 1

Marketing ansvarar för medlemsutveckling, samordning med andra organ, utåtriktade kontakter i största allmänhet m m.

En specifik sådan utåtriktad manifestation är den av OMG anordnade konferensen/utställningen ObjectWorld. Mer om denna aktivitet under avsnitt 5.4 nedan.

Program Management är ett nyinrättat ansvarsområde med primär uppgift att intensifiera samarbetet med etablerade standardiseringsgrupper inom objektorientering samt med olika industrigrupperingar. Syftet är givetvis att få dessa att acceptera eller åtminstone att titta närmare på OMGs resultat. Program Management har också på sin lott att etablera samarbete med slutanvändare, d v s de organisationer och personer som avser att köpa in respektive använda produkter baserade på OMGs standarder.

Det direkta utvecklingsarbetet bedrivs inom Technical Committee. Den är organiserad enligt figur 2. Mer om arbetets organisation under avsnitt 3.



Figur 2



En Subcommittee är en permanent grupp med syfte att ge allmänna rekommendationer till TC inom sitt ansvarsområde.

Special Interest Groups (SIG), å andra sidan, kan vara av mer temporär natur. De uppstår på initiativ av personer som vill titta närmare på någon speciell OO-aspekt genom diskussionspapper, analyser, enkäter m m. En för närvarande mycket aktiv SIG är Analysis and Design. Gruppen har tittat på och jämfört olika analys- och designmetoder, kommer att gå igenom existerande Case-verktyg som stödjer OO-baserad utveckling och kommer också att försöka definiera en generell super-OO-modell bland annat.

Arbeten av primärt intresse för OMG bedrivs inom Task Forces (TF). En TF har att, inom sitt täckningsområde, avgränsa viktiga aspekter för standardisering, begära in förslag från industrin och att utvärdera dessa.

## 2.3 Synpunkter

Makten inom OMG ligger mycket klart hos corporate members. Dessa har vardera en röst, arbetar nära den fasta OMG-personalen och kan delta i alla aktiviteter. Associate members saknar rösträtt men kan delta i och därmed påverka det konkreta arbetet. Corporate och associative members är främst leverantörer med ett marknadsintresse inom objektorientering (OO).

End-user members ses som en enhet eller Special Interest Group (SIG) där gruppen som helhet har en röst. Som framgår av namnet består denna grupp främst av kunder till OO-produkter i vid mening.

University och subscribing members har tillgång till all producerad dokumentation.

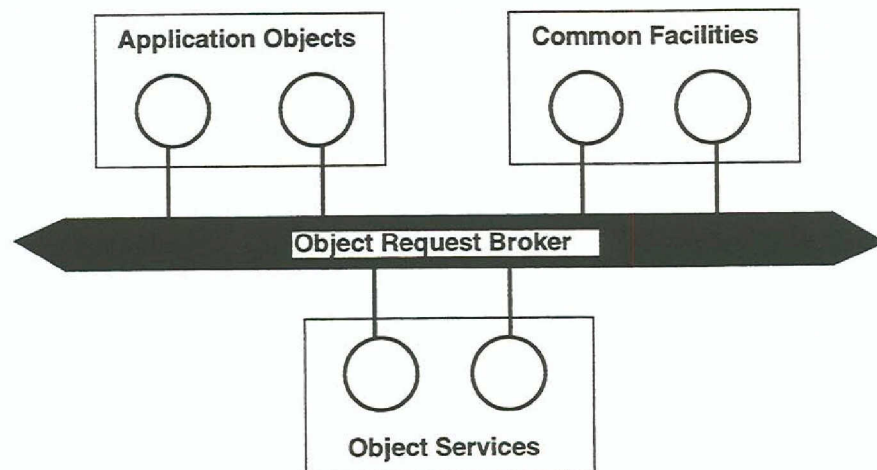
# 3. Standardiserings- arbetet

## 3.1 Vad är av intresse för standardisering?

Som bas för OMGs arbetsinriktning ligger en definierad referensmodell över Object Management Architecture (OMA). Grundad på OMGs primära syfte (se avsnitt 1, ovan) formulerar referensmodellen en vision över hur tillämpningar av olika slag bör samverka i en distribuerad miljö med hjälp av ett objekt-orienterat synsätt.

Referensmodellen beskriver i allmänna termer de egenskaper som en OMA ska uppfylla. Den bildar på så vis ett ramverk inom vilket lösningar på olika delproblem inom ett OMA kan hänföras. Ramverket definierar till exempel i övergripande termer de, enligt OMGs perspektiv, olika beståndsdelarna i ett OMA, vilka egenskaper de ska uppfylla och hur de ska samverka (protokoll, restriktioner m m). Se figur 3. Inom ramverkets ram finns utrymme för många olika lösningar.

Standardiseringsprocessen bedömer och bearbetar olika förslag till lösning.



Figur 3

De företagsverksamheter som har behov av att samarbeta med andra företagsverksamheter går under den gemensamma benämningen objekt (object). Ett objekt är en programmodul, alltifrån en stor fristående tillämpning, exempelvis en kundreskontra ner till en liten "kodsnutt", eller en sorteringsalgoritm. Gemensamt för dem är att de antingen behöver ta hjälp av andra objekt eller är villiga att ge hjälp till andra objekt. Ofta gäller både och. Objekt kan i princip även vara data. Enligt

objektorienterat synsätt ställer objekten sin hjälp till förfogande genom en uppsättning tillgängliga egenskaper och beteenden. Hur dessa i praktiken realiserar ligger dolt (inkapslat) i varje objekt.

Den tillgängliga servicen efterfrågas av den som behöver den (client) genom att en begäran (request), d v s ett meddelande placeras hos en meddelandeöverförare, ett slags postverk, på vars ansvar det ligger att hitta leveransadress, att leverera meddelandet ifråga dit, samt att returnera eventuella svar. I OMGs termer heter denna funktion Object Request Broker (ORB).

OMA grupperar objekten i tre undergrupper:

- Application Objects

Där återfinns en organisations egna tillämpningar eller delmoduler, skapade och avgränsade i enlighet med organisationens egna behov.

- Common Facilities

Hit hör funktioner som kan förväntas vara av intresse inom många tillämpningar och av den anledningen förtjänar en exakt specifikation avseende funktionalitet och gränssnitt. Sannolikt är det rutiner som man köper istället för att utveckla själv. Det kan i princip finnas många leverantörer av samma Common Facility. Gränslinjen mellan ett egenutvecklat Application Object som används för många syften inom en organisation och en Common Facility är något diffus. Exempel som brukar nämnas är hjälpfunktioner, utskriftsfunktioner, e-mail, frågefunktioner, ...

- Object Services

Dessa har ingenting direkt med tillämpningens funktionalitet att göra. Snarare är de att ses som systemfunktioner för att hantera objekt. Objekt ska kunna skapas och raderas. De ska kunna skyddas mot obehörig användning. En tillämpningsoperation kanske involverar ett antal objekt med givna beroendeförhållanden dem emellan. Sådana sammansatta operationer kan behöva regleras genom transaktionsmekanismer. Objekt är instanser av klasser och ska bete sig i överensstämmelse med vad som stipuleras för klassen. Klasser eller ett objektschema måste kunna hanteras.

Likheten med vad som normalt återfinns hos databashanteringssystem är påfallande.

För samtliga tre kategorier gäller att de behöver samverka. Samverkan sker lämpligen över en ORB.

OMG är på sikt intresserade av standardisering av OMAs alla delar utom givetvis Application Objects. Som framgår av figur 2 finns än så länge endast TFs för ORB och Object Services. Arbete inom Common Facilities anstår tills vidare.



### 3.2 Standardiseringsprocessen i stort

TC träffas ca 5-6 gånger per år. Mötena läggs på lämplig plats i världen, ofta i USA. Mötet under juli 1993 var förlagt till Chicago. Stockholm (Ericsson) var värd i september 1993. December 1993 kommer turen till Austin, Texas, o s v. TFs möten kan sammanfalla med TC, men kan också mötas oftare vid behov. SIGs har ingenting med det direkta standardiseringsarbetet att göra. Av praktiska skäl sammanfaller dock normalt dessas möten med TC.

Jämfört med standardiseringstempot inom ISO måste man säga att OMG arbetar med en föredömlig snabbhet.

Vilka delmoment genomlöper då en specifikation på vägen mot OMG-standard?

En TF ansvarar för sin del av referensmodellen. När den finner behov av kompletteringar inom ännu inte täckta delar av OMA kan den välja att först "syna" intresset och den aktuella marknaden genom att skicka ut en förfrågan till industrin om deras planer eller produkter inom aktuellt område. Detta kallas Request for Information (RFI).

När TF bedömer tiden mogen att driva fram en viss standard sänds en Request for Proposals (RFP) ut. En RFP kan rikta sig till alla intresserade eller till endast vissa organisationer (exempelvis bara Corporate Members). Inkomna förslag kallas för Proposed Technology. Dessa presenteras för TC varefter bedömningssarbetet delegeras till aktuell TF. TF levererar så småningom sin rekommendation till TC för omröstning. Går förslaget igenom i TC skickas rekommendationen vidare till BOD för slutlig acceptans som Sponsored Technology eller, i vardagligt tal, som en standard.

Intressant och ganska unikt i standardiserings-sammanhang är att ingenting accepteras som en Sponsored Technology om det inte finns implementerat i ett fungerande skick. Dessutom gäller att endast Corporate Members får vara förslagsställare. Om en utomstående organisation lämnar in ett svar på ett RFP och förslaget senare accepteras, måste antingen organisationen själv bli en Corporate Member eller licensiera teknologin till en sådan.

Principen undanröjer risken för "pappersdrakar", tillåter en snabb process, där underlaget är klart och endast mindre modifieringar kan bli aktuella, samt lägger själva utvecklingsarbetet på företagens axlar snarare än som en process inom OMG.

### 3.3 Synpunkter

- Man ska hålla i minnet att OMG inte naturligt tillvaratar hela objekt-orienteringsområdets angelägenheter. Röstreglerna gör OMG i praktiken till en leverantörsorganisation för att effektivt tillvarata deras intressen. Det är knappast automatiskt givet att dessa alltid sammanfaller med dem som användarna kan bedömas ha. De senare borde antingen företrädas av en egen organisation eller ges ett ökat inflytande inom OMG. I ett övergripande perspektiv borde fokus läggas på att lösa verkliga problem, dvs kundens problem, snarare än OO-produktleverantörens.

Å andra sidan förutsätter detta att kunderna dels artikulerar sina behov, dels kan sätta tillräcklig kraft bakom kraven. Vid OMG-mötet i juli 1993 uttrycktes en klart ökad irritation över den aktuella slagsidan i makt-hänseende inom OMG. Ett resultat av mötet blev en deklaration bland närvarande "kundmedlemmar" att vitalisera deras End-User Requirements Special Interest Group (se figur 2). OMG tycks ha tagit intryck av dessa strömningar genom att ge det nyinrättade Program Management i uppgift att aktivt sköta kontaktytan mot de så kallade slutanvändarna.

- Ett utslag av maktsituationen är också ett tassande runt reella problem, med syftet att föreslå standarder som inte "trampar någon leverantör på tårna". Vilken leverantör vill vara med och betala i en sammanslutning, som kan komma att föreslå något som ligger helt vid sidan om, eller värre, strider mot egna investeringar inom området? Å andra sidan kan man ju vara säker på att godkända standarder är mycket väl underbyggda, oftast genom redan existerande produkter. Se vidare nästa avsnitt.
- Mycket grovt kan OMGs roll ses som en intresseorganisation som idésamordnar corporate members intressen inom OO på dessas villkor samt sprider resultaten för övriga att anpassa sig efter. Man etablerar ett slags de facto standarder i och med att corporate members inkluderar de största aktörerna inom OO-området.
- Standardiseringsprocessen gynnar den stora organisationen, dels genom att den är en viktig medlem i OMG att ta hänsyn till, dels att den har större resurser att ta fram förslag baserade på redan existerande produkter eller långt framskridna utvecklingsplaner. Å andra sidan gör man heller ingen hemlighet av detta.
- Arbetet på en standard utförs alltså primärt inom en organisation eller inom en grupp av organisationer. OMGs möten utgör mer ett diskussionsforum kring färdiga förslag.



# 4. Framtagna "standarder"

Som bas för alla specifikationer ligger en enhetlig objektmodell kallad Object Model. Den formulerar den abstrakta uppfattning och de regler som ska genomsyra alla konkreta objektmodeller, som formuleras i olika standard-specifikationer. Object Model beskrivs översiktligt i avsnitt 4.1. Den först accepterade Sponsored Technology, som dessutom genomlöpte hela arbetsförloppet var en specifikation av en ORB med titeln "Common Object Request Broker Architecture", förkortad CORBA 1.1. Denna ORB beskrivs i avsnitt 4.2.

## 4.1 Object Model

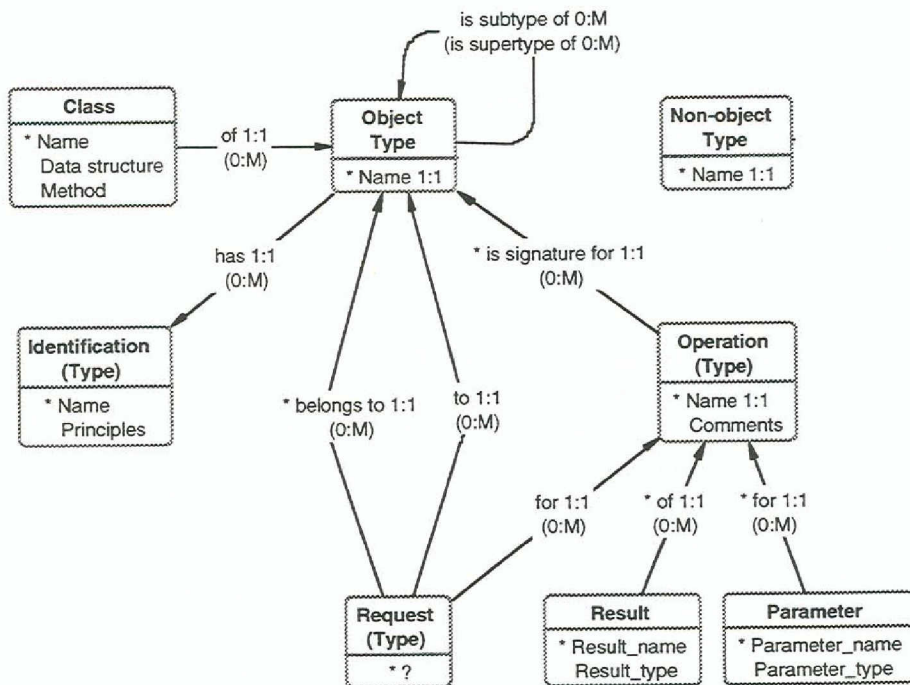
### 4.1.1 Inledning

Object model består av en uppsättning väl beskrivna, implementeringsoberoende begrepp att användas i samband med formulering av standard-specifikationer. Syftet är att lägga en stabil semantisk plattform i form av väldefinierade begrepp. En objektmodell i vanlig bemärkelse kan man därför knappast tala om. Begreppen refererar dessutom till företeelser på flera modellnivåer. I termer av IRDS Framework, ISO 10027, syftar begreppen både till företeelser på IRD Level och IRD Definition Level.

Uppdelning har gjorts i en fundamental del som alla standarder ska svara upp mot och en tilläggsdel. Den fundamentala kallas för Core Object Model. Den som så önskar tillför egna begrepp som *components*. Vissa uppsättningar *components* är användbara för ett visst ändamål, andra för ett annat ändamål. Med hjälp av konstruktionen *profile* kan *components* grupperas med tanke på visst syfte. Tills vidare hoppas man att Object Model ska vara den stabila begreppsbasen vid formulering av specifikationer och mekanismer så att alla med kunskap om modellen ska ha en rimlig chans att uttolka samma semantik ur beskrivningen. I OMGs termer kallas det "design portability".

### 4.1.2 Modellen

Figur 4 är ett försök till begreppsmodell av Core Object Model. Den baseras på kapitel 4 i skriften "Object Management Architecture Guide". Eftersom definitionerna ibland är otydliga kan fel ha smugit sig in. I vissa fall innehåller figuren preciseringar av exempelvis sambandsegenskaper som är rena tolkningar av löpande text.



Figur 4

Type inom parentes är tillägg för att förtydliga innebörden, men som OMG valt att inte inkludera i sina benämningar.

Centralt i modellen ligger Object Type. Den karakteriseras av en implementering Class, en princip för Identifiering och gränssnittet mot omvärlden, d v s alla tillgängliga Operation (Type)s. Varje Operation (Type) beskrivs med sitt namn och Object type-tillhörighet. Därutöver kan parametrar behövas som en nödvändig förutsättning för operationens genomförande, samt eventuellt en uppsättning resultatuppgifter. En Request är en begäran om att få en viss Operation utförd. En Request (Type) pekar alltså ut vilken typ av Operation det är fråga om, varifrån begäran kommer (belongs to) och för fullständighetens skull vilken Object Type som operationen anses tillhöra. Den sistnämnda uppgiften kan i normalfallet härledas ur aktuell Operation, dock inte om samma Operation återfinns på flera platser i subtypstruktur och den lokala inte efterfrågas. Subtypsstrukturer med multipla supertyper kan definieras. Ibland finns det behov av att beskriva företeelser som inte svarar mot kravet på att vara objekt. Dylika går under beteckningen Non-Objects.

Utöver begreppen på denna modellnivå konstaterar man behovet av unika interna identifierare för varje objekt (OID). Hur dessa implementeras, ligger utanför modellens ansvarsområde att reglera.

Ett antal fördefinierade Operation types nämns också. Dit hör ett antal mängdoperationer, exempelvis Obj som ger mängden av alla OIDs och Nobj som ger alla non-objects.

Samtidigt definieras generella mängdoperationer (egentligen på nästa abstraktionsnivå) av typ OTypes (alla object types) och NTypes (alla Non-object Types).

Anledningen till att just dessa mängdoperationer preciseras är oklar.

#### **4.1.3 Synpunkter på Object Model**

Object Model fyller ett första syfte som semantisk bas, men kommer med största sannolikhet att behöva preciseras framöver, inte minst när den heterogena miljön kommer att behöva beskrivas mer formellt, exempelvis i någon form av repository. Då kommer också information mellan repositoryn att behöva utbytas. En heterogen, samverkande miljö i enlighet med OMGs referensmodell, kan ju i vid mening ses som ett applikationsområde beskrivbart i en applikationsmodell (IRD Level) i termer av begrepp definierade i en begrepps- eller metamodel (IRD Definition Level). Detta klarar man inom många andra applikationsområden. Också inom det mycket närliggande PCTE, vilken definierar interoperabilitet mellan fristående tillämpningar inom området datorstött mjukvaruutveckling, har man en formellt definierad metamodel. Den enda skillnaden är att den inte är rent objektorienterad i dynamisk bemärkelse.

Det är oklart varför inte OMG på motsvarande sätt försöker formulera en objektbaserad metamodel. Överhuvudtaget känns kapitlet ofärdigt. Det rör snarare till uppfattningarna hos en redan OO-kunnig person, än reder ut dem.

Att notera är avsaknaden av motsvarighet till de i vanliga datamodeller förekommande begreppen attribute type och relationship type. Å andra sidan ingår inte relationship types i den renodlade OO-filosofin.

Oklarheten kring den abstrakta objektmodellens roll ökar vid jämförelse med den konkreta objektmodell CORBA använder sig av. Tanken är att de konkreta modellerna ska vara preciseringar av den abstrakta modellen med hänsyn till implementeringstekniska eller andra förutsättningar. I realiteten använder CORBA-modellen ibland andra begrepp (exempelvis object reference isället för identifier).

Subcommittee för Object Model har varit passiv ett tag, men krafter är igång för att försöka aktivera den igen. Det finns jobb kvar att utföra!



## 4.2 Common Object Request Broker Architecture 1.1 (CORBA 1.1)

### 4.2.1 Bakgrund

CORBA 1.1-specifikationen är en kompromiss mellan två förslag. Det ena förslaget inlämnades gemensamt av Hewlett-Packard, NCR, ObjectDesign och SunSoft. Det bygger på statiska, förbestämda förutsättningar. Det andra kom från Digital och HyperDesk och bygger istället på mer dynamiska förutsättningar, d v s man vet inte i förväg vilka requests som kan komma och inte till vilka objekt.

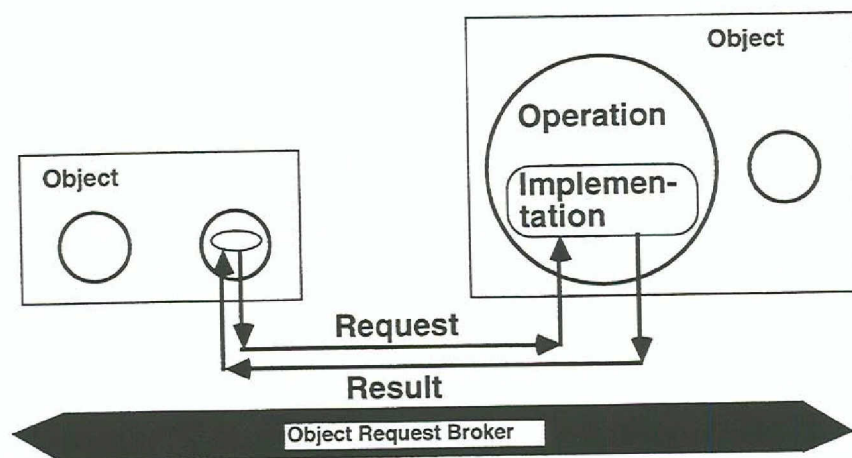
Förslagen hade vart och ett sina förtjänster. De representerade helt enkelt två olika principlösningar. Den statiska gav prestanda, den dynamiska flexibilitet. Istället för att tvingas välja ett av dem, återremitterades förslagen till de båda grupperna med en önskan om ett kompromissande till ett gemensamt förslag. Tillbaka kom i princip en lösning (Proposed Technology) som innefattade summan av de båda förslagens ingredienser. Förslaget bearbetades, inte förvärande, inom Object Request Broker Task Force (ORBTF), se figur 2. I september 1991 godkändes det som en Sponsored Technology.

### 4.2.2 Översikt, ORB

En Object Request Broker (ORB) är ett slags nervsystem eller meddelandehanteringssystem som gör det möjligt för objekt att "prata" med varandra i en distribuerad objektmiljö. Mer specifikt är en ORB en mekanism som för någons (clients) räkning

- tar hand om ett request, innehållande en önskad objekt-operation och eventuellt några parametrar, samt noterar vem avsändaren är
- kontrollerar att aktuellt request är formellt korrekt
- letar upp en användbar implementering av aktuell operation
- skickar över request inklusive de eventuella parametrarna
- begär att avsedd operation utförs
- lämnar tillbaka svar till klienten, när så är aktuellt.

Se figur 5. Värt att notera är att det kan finnas flera implementeringar av samma operation. Ett request formuleras bara i termer av en abstrakt objektmodell.



Figur 5

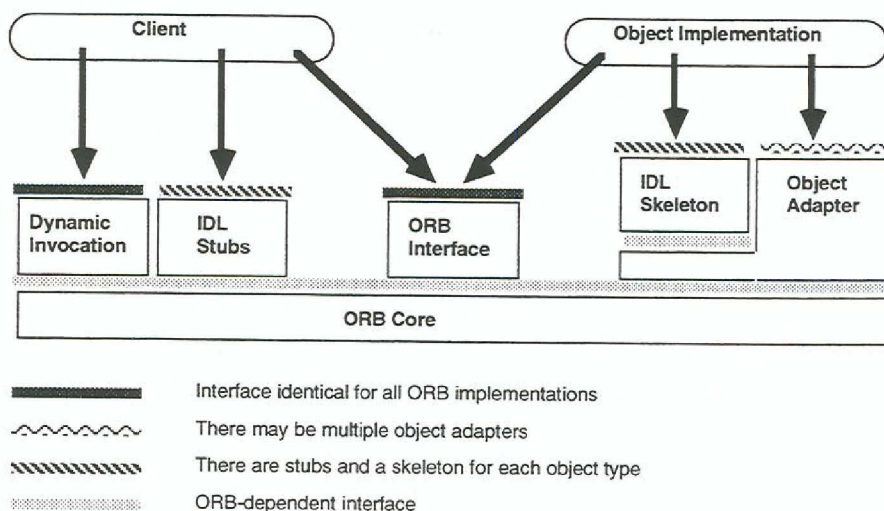
Det hela låter enkelt men är i realiteten ganska komplext. Över till den lösning CORBA 1.1 anvisar. Observera först att CORBA inte är en implementering gjord av de sex företag som lämnade in förslaget, utan en specifikation som kan realiseras i form av en mängd olika implementeringar. Däremot är standarden baserad på erfarenheter från existerande produkter/prototyper hos dessa företag, enligt reglerna för acceptans av en standard.

#### 4.2.3 Översikt över CORBA 1.1

Avsikten med detta avsnitt är endast att mycket kort beröra några av de mer påfallande ingredienserna i standarden. För en fullständig beskrivning hänvisas till dokumentet "The Common Object Request Broker: Architecture and Specification" utgiven av OMG och X/Open. För den allmänt intresserade är det knappast väl använd tid att gå in i standardens alla detaljer. För denne är visionen det intressanta, inte alla mer eller mindre komplexa tekniska överväganden och delresultat på vägen dit. CORBA 1.1 innehåller "våldigt många träd i en ganska diffus skog". Den existerande standarden är också en första specifikation. Nästa version (CORBA 2.0) har lämnats ut som en RFP. Fler lär så småningom följa. Visionen är enkel och klar, att förverkliga den är synnerligen komplext!

Figur 6 är en ofta visad skiss över de olika involverade gränssnitten i samband med ett request.





Figur 6

Med risk för missstolkningar av standarden följer här några kommentarer till bilden:

*Dynamic Invocation* är en generell biblioteksrutin för en client. Den anropas i det dynamiska fallet, d v s när man inte kan förutse vilken objekt-operation som kan behöva exekveras. Gränssnittet är standardiserat i CORBA vilket gör att samma formulering av en begäran kan användas oavett vilken ORB-implementering man nyttjar. Det specificerar den syntaktiska och semantiska uppbyggnaden av ett request (referens till ett objekt, vilken operation det är fråga om och en lista på parametrar). Sedan återstår att kontrollera levererad request, ta reda på var en lämplig implementering kan finnas o s v.

*Object Adapter* står för en uppsättning generella services som en viss implementering av ett objekt (och dess operationer) behöver för att introducera sig för ett ORB. Det behövs flera sorters object adapters beroende på objektens allmänna användning, karakteristik o s v. En object adapter innehåller också stöd för att aktivera/deaktivera de methods som inom ett objekt implementerar en viss operation, ombesörjer viss behörighet m m. Dessa egenskaper behövs, inte minst för att klara av de dynamiska anropen. *Dynamic Invocation* och *Object Adapter* kan grovt ses som komponenterna i den dynamiska ansatsen.

En *IDL Stub* är en biblioteksrutin. Det behövs en stub per typ av anropat objekt. Objekttypen definieras med hjälp av det implementeringsoberoende, i CORBA definierade språket *IDL* (Interface Definition Language). Stuben kompileras och länkas in i clientprogrammet och erbjuder därigenom en direkt länk till motsvarande implementering. Dess motsvarighet på implementeringssidan är ett *IDL Skeleton*. Stub och Skeleton utgör tillsammans den statiska ansatsen. Anropen blir här maximalt effektiva.

*ORB Interface* svarar mot några standardiserade funktioner.

Moderna tillämpningar kommer att bli alltmer komplexa och ha allt större behov av att etablera kontaktytor med omgivande system. Tillämpningarna kommer i allt större utsträckning att bygga på återanvändning av objekt (moduler). Allt detta talar för att den dynamiska ansatsen kommer att få ökad relevans. I samband därmed kommer också behovet av en detaljerad beskrivning av existerande objekt att öka, deras karakteristik och krav, var de finns, kanske vad de kostar o s v. OMG talar här om ett *Interface Repository*. Ett gränssnitt finns definierat inom CORBA, men man kan förvänta sig att en hel del kompletterande arbete kommer att initieras framöver inte minst vad gäller lämpligt innehåll. En RFP har nyligen gått ut med syfte att få in förslag som med större fullständighet och precision definierar ett Interface Repository.

Den dynamiska ansatsen kommer också att ställa högre krav på intelligens i ORB, exempelvis för att söka den implementering som för tillfälligt kan utföra jobbet effektivast (ligger kanske på en lågbelastad nod), för att dynamiskt sprida och kanske duplicera implementeringar med hänsyn till något optimeringskriterium o s v.

För övrigt bör påpekas att ordet objekt för O i CORBA inte alls innebär krav på att objekten är implementerade i något objektorienterat språk. Objekt står för att uppträdandet ska vara objektmässigt d v s att de ska kunna sända och acceptera CORBA-uppbyggda requests. CORBA skulle vara av synnerligen begränsat värde om det utestängde alla befintliga, konventionellt implementerade tillämpningar (legacy systems). För icke-OO tillämpningar kompletterar man med ett skikt som översätter mellan OO-formuleringar och tillämpningens lokala sätt "att se på världen".

#### **4.2.4 Synpunkter**

CORBA 1.1 ger en känsla av en implementeringsmässig kompromiss. Hur långlivad den blir kommer att bero på om kommande versioner kommer att erbjuda kompatibilitet eller ej. Det intressanta är grundidéerna på ett konceptuellt plan. Man vill ju helt enkelt kunna skicka meddelanden mellan objekt oavsett deras placering. För det behövs ett name space och om det finns flera konkurrerande (exempelvis ett per ORB) behövs en intern brygga mellan dem. Objektmodellen kunde lämpligen beskrivas i en schemadefinierad objektbeskrivningsbas (repository+schema), som gärna får vara distribuerad. Filosofin bakom PCTE känns mer genomtänkt och elegantare. Å andra sidan kan den ansatsen istället komma att erbjuda prestandaproblem. Det snabbt ökande marknadsstrycket och skickligheten att manövrera detta, kommer ganska snart att utkristallisera de mest bärkraftiga de facto-standarderna.

För övrigt kan nämnas att OMG nyligen inrättat en ny SIG för att "titta närmare på" PCTE.



# 5. Pågående arbete

Som synes av figur 2 finns för närvarande två Task Forces, en med inriktning mot Object Services (OSTF), en med inriktning mot vidareutveckling av CORBA-specifikationen. CORBA (ORBTF).

## 5.1 Arbete inom OSTF

En Object Service (OS) är en generell funktion med en tänkt användbarhet i många olika situationer. Object services utnyttjar normalt egenskaperna hos CORBA för att utföra sina specifika arbetsuppgifter. En OS kan, om man anlägger en skiktindelad syn, anses ligga på en nivå över CORBA. Listan över aktuella OS har tagits med, mer för att visa vilka frågeställningar man har att brottas med i anslutning till en ORB, än att tala om vad som just pågår inom OSTF. Bland aktuella OS återfinns

- Lifecycle, som reglerar hur man skapar, raderar och kopierar objekt.
- Naming, som reglerar hur objekt ska namnges för att tveksamhet om unikheter vid referenser inte ska kunna uppstå.
- Event notification, ombesörjer information om inträffade händelser till de objekt som önskar känna till det.
- Persistence, som reglerar hur objekt hanteras när de ska ha ett mer permanent liv.

De tre första accepterades av Task Force vid juli-mötet. Det har därefter godkänts av TC. Om de till sist godkänns av OMGs BoD blir de den andra omgången Sponsored Technology efter CORBA 1.1. För Persistence föreligger flera förslag som återremitterats för sammanjämkning till ett förslag.

En ny RFP (RFP 2) avlämnades i juli 1993 med inriktning mot services av den typ man ofta återfinner i samband med databashanteringssystem. Bland dessa återfinns

- Transaction; hur integriteten upprätthålls hos objekts data.
- Concurrency control; hur samtidiga anrop till ett och samma object ska regleras.
- Time; hur tid vid behov ska kunna synkroniseras över ett nätverk.
- Externalization; extracting an objects internal state (vad det nu betyder).
- Relationships; hur man bör hantera samband mellan olika objekt.

Deadline för förslag var november 1993. Därefter kommer bearbetning förhoppningsvis att utmynna i förslag som kan accepteras av TF under senare delen av 1994.

Likheten i krav jämfört med hantering av objekt eller entiteter i en databas är påfallande och ganska naturlig. I båda fallen är det ju fråga om att komma åt objekt utan att veta var de finns och att på något sätt, för något behov, titta på eller påverka deras tillstånd (eller beteende). Frågan är vari skillnaden ligger mellan en applikationsmiljö under stöd av CORBA och en där applikationen ligger som objekt i en helt distribuerad OO-databas av en typ som inte är språkberoende?

## 5.2 Arbete inom ORBTF

Version 1.1 av CORBA är en accepterad OMG-standard. Efter dess tillkomst hösten 1991, har förslag och önskemål om en utvidgning förts fram av många parter. Dessa idéer kanaliserades genom en RFI under december 1992 som utmynnade i över 30 problemområden att arbeta vidare med. Till dem hör behovet av att samarbeta mellan olika ORBs, inte bara inom ett ORB. Andra behov berör hantering av ickesynkroniserad meddelandesändning. (Man skickar ut en request och fortsätter jobba. I en del fall vill man så småningom få reda på hur det gick, i andra saknar det betydelse.) Det börjar även efterfrågas anpassning till andra delvis överlappande standardiseringsansatser. Till dessa hör ISOs ODP och X/Open.

Diskussionen har ringat in de viktigaste behoven och utmynnade i två RFP. Den ena går under arbetsnamnet *Initialization/Interoperability* och innehåller just problematiken runt samverkan mellan flera ORBs. Man bör kunna placera sina program och data utan att behöva ta hänsyn till underliggande hårdvara, mjukvara eller ORB. Initieringsdelen omfattar problem i samband med öppning av kommunikation mellan ett client object, ORB och services. Detta täcktes inte in tillfredsställande i CORBA 1.1. Ofullständigt behandlat var även det tidigare nämnda *Interface Repository*. En mer omfattande specifikation begärs i det andra RFP. Deadlines för svar är mars respektive maj 1994.

Inom ORBTFs område ligger också problematiken kring avbildning mellan OMGs gränssnittsspråk IDL och de språk som tillämpningarna normalt är skrivna i. En standard underlättar tillämpningens flyttbarhet mellan olika miljöer. CORBA 1.1 innehåller en mappning mot C. I en andra omgång har ett RFP gått ut med avseende på C++. Två förslag har inkommit. Acceptans förväntas i början av 1994. Andra språk kommer att följa. På sikt bör avbildningar finnas mot samtliga språk som har en rimligt bred användning, inte bara objektorienterade språk.

På sikt kan man också förvänta sig än mer automatik, optimering och service från en ORB. Exempelvis kan det finnas flera objekt som kan utföra en viss begäran. ORB skulle då kunna välja ut den lämpligaste med hänsyn till olika kriterier.



Istället för att bara skicka meddelanden mellan två objekt skulle dess roll kunna bli en aktiv övervakare och styrare över en total objektmassa. Kanske behöver man, för att uppnå detta, ta ett steg tillbaka från standardisering av existerande teknik och angripa problemet förutsättningslöst (och antagligen med en god portion vision).

### 5.3 Common Facilities

Detta är ett område av arkitekturen som man inte ännu orkat ta itu med.

Exempel på tänkbara facilities att "styra upp" är

- Help facility
- Error reporting
- Printing and spooling
- Electronic mail facility
- Query facility
- Computer-based training

### 5.4 Löpande

Sannolikt utgör OMG-konferensen Object World den viktigaste plattformen för att föra ut det objektorienterade budskapet. Den första hölls 1991 i San Francisco. Förutom en årlig upprepning i San Francisco, "turnerar" numer konferensen till olika hörn av världen. Under hösten 1993 återfinns den i Australien (september), Tyskland (september), Storbritannien (oktober) och Japan (oktober). I januari 1994 var det Bostons tur och i juli 1994 San Franciscos tur igen. Med denna intensitet sprids självfallet budskapet på bred och snabb front. Frågan är om denna strategi är lämplig när väl det initiala intresset har stillats. Händer det så mycket nytt, så snabbt att inte återkommande besökare tröttnar? I alla händelser utgör den en allvarlig konkurrent till den tidigare allenarådande, årliga konferensen OOPSLA.

En rollfördelning tycks ha uppstått mellan de båda konferenserna såtillvida att ObjectWorld vänder sig till affärsfolk med ett intresse i att köpa produkter eller på annat sätt införskaffa konkret kunskap med direkt koppling till behov i den egna organisationen, medan OOPSLA antagit en mer akademisk profil. ObjectWorld försöker täcka in objektorienteringens alla upptänkliga områden medan OOPSLA av tradition varit relativt språkorienterad.

ObjectWorld expanderar både vad gäller besökare (ca 5000 i San Francisco 1993) och utställare.

Man ska komma ihåg att föredragen på ObjectWorld inte är föremål för samma urvalsprocess som vanliga, mer akademiska konferenser. Proceedings består i allmänhet av ljusbilder. Presentatörer har en tendens att dyka upp år efter år.

Syftet är med andra ord i första hand att marknadsföra och informera om objektorientering, inte nödvändigtvis att föra fram nya, intressanta landvinningar.

OMG ger också ut tidskriften *First Class* med 6 nummer per år. Den innehåller kortare artiklar, referat från konferenser, annonseringar om kommande aktiviteter, nya medlemmar m m. Årsprenumeration kostar ca 50 dollar. Se sista sidan för adress.

# 6. Milstolpar

Hittillsvarande OMG-aktiviteter kan sammanfattningsvis formuleras genom följande milstolpar:

- 1989 Bildades OMG
- 1990 Object Management Architecture Guide, version 1.0
- 1990 RFP avseende ORB
- 1991 Första Object World konferensen
- 1991 CORBA 1.1, Sponsored Technology
- 1992 Core Object Model specificerad
- 1992 Andra Object World konferensen
- 1992 Object Management Architecture Guide, version 2.0
- 1992 RFP avseende Object Services
- 1992 RFP avseende C++ bindning
- 1993 Tredje årliga Object World konferensen
- 1993 RFP avseende fler Object Services
- 1993 RFP avseende CORBA 2.0



# 7. Avslutande synpunkter

Existerande OMG-standarder är bara en början på en lösning av problemen med samverkan mellan komponenter i distribuerade tillämpningar. Dessutom är inte standarderna speciellt objektorienterade eller åtminstone inte helt präglade av ett objektorienterat tänkande. OMGs VD Chris Stone lär också ha sagt vid en presentation nyligen att om OMG skulle starta idag skulle gruppen inte heta OMG utan "Distributed Applications Group". Samverkan är nyckelordet. Sannolikt kommer det att ta lång tid innan en fulltäckande eller sammanvävd OMA finns tillgänglig.

Å andra sidan finns det snart tillräckligt mycket för leverantörer att bygga på, vilket i sin tur gör att erfarenheter snart kommer att finnas, som i sin tur kommer att generera nya uppslag och revideringar o s v. Området är sannolikt så komplext att det vore en omöjlighet att bygga en full och hållbar standard utifrån enbart teoretiska resonemang. Trial-and-error strategi kan i vissa lägen vara nog så produktiv.

OMG ser sig idag som det enda reella standardiseringsorganet i den objektorienterade världen. Det är sant att man arbetar med stor intensitet och med många (dock i stor utsträckning passivt nyfikna) företag som medlemmar. Just snabbheten är den inneboende styrkan. Vad man inte får glömma är att det parallellt pågår mycket seriöst och delvis överlappande arbete inom andra standardiseringsinstanser, inte minst inom ISO. Dessa kan ha en något annorlunda inriktning, baserad på en långsiktig kontinuitet (exempelvis ODP och nya IRDS-ansatser). Alternativt kan syftet vara specifika applikationsområden (exempelvis GDMO) eller helt enkelt behovet av en annan infallsvinkel på begreppet objektorientering (exempelvis ANSI/X3H7). På sikt måste OMG ta hänsyn till dessa "kraftfält" för att behålla sin trovärdighet. Hur kommer man att fungera i en dylik kompromissinriktad roll jämfört med den existerande inställningen, något grovt formulerad som "låt den starke vinna"?

Antagligen fullgör OMG idag en viktig roll. Det gäller att ta fram verktyg som bygger på en gemensam filosofi eller åtminstone befinner sig inom drivkrafter mot en framtida idégemenskap och interoperabilitet. På sikt gäller det däremot att försöka ta ett steg bort från produktinriktningen och försöka lägga ett idémässigt stabilt ramverk kring objektorientering. Området är fortfarande vildvuxet, men inom allt fler applikationsområden (databaser, systemutveckling, Case, metoder, telekommunikation, distribuerade system, ....). Dessa kanske lever sina fristående liv idag men kommer med största sannolikhet att behöva samarbeta i morgon. De bör åtminstone ges möjlighet att lära av varandra, att få stöd och råd. Att uppfinna hjulet på nytt för varje applikationsområde, dessutom med ibland högst tveksamma resultat (p g a höggradigt

tillämpningsfokus eller att man kastar sig över den nya "undermedicinen" OO och i avsaknad av metod/modellkunskap skapar någon halvmesyr), är inte produktivt.

Långsiktigt bärkraftig standard, oavsett om den består av referensmodeller, exakta gränssnittoperationer eller dylikt bör tas fram inom organ som är partsobundna. Den naturliga samlingsplatsen är givetvis de nationella standardiseringsorganen, ISO m fl. Den oftast inbyggda trögheten hos dessa motiverar givetvis OMGs och andra liknande organs mera effektiva arbete. Men det ska alltid finnas en varudeklaration beträffande deras roll och drivkrafter. De utför ett ambitiöst arbete, tillåter sig att söka, testa, misslyckas och når därför så småningom fram till en mycket seriöst genomarbetad partsinlaga.

OMG motverkar en totalt vildvuxen utveckling genom att inordna de kommersiella "krafterna" under ordnade former. Det är ju inget som sedan hindrar ANSI och ISO att driva samma eller överlappande frågor i ett formellt standardiseringsarbete. Många anser dock att de ofta kommer för sent, vilket som sagt, snarare får tillskrivas dessa organs inneboende tröghet, än riktas som kritik mot OMG.

Visar sig OMGs standarder, utifrån ett helhetsperspektiv, ligga fel kommer ju ändå formella standarder att så småningom tränga ut OMGs motsvarighet och ge OMG dålig reklam. Detta är givetvis OMG mycket medvetna om och försöker undvika genom en seriös process. Skulle OMG någon gång komma att uppfattas som ett organ för ett antal starka aktörer på marknaden (vilket det ju lite elakt är fråga om), skulle snart deras goda rykte och medlemstal minska.

På tal om medlemmar; 340 medlemmar betyder inte med nödvändighet ett otroligt starkt stöd för OMGs arbete. Det kan vara ett tecken på ett mycket stort intresse för och behov av att "hålla ögonen på" vad som händer inom OO. Några andra OO-organ med internationell räckvidd, förutom de mer "oåtkomliga" formella standardiseringsorganen finns ännu inte.

En bred acceptans av OO, baserad på klart definierade specifikationer och definitioner, är en förutsättning för områdets långsiktiga överlevnad.

## **Litteratur utgiven av OMG**

Object Management Architecture Guide, andra upplagan 1992

The Common Object Request Broker: Architecture and Specification, 1991

## **Adress till OMG**

492 Old Connecticut Path  
Framingham, MA 01701  
USA  
Tel: 0091-(508) 820 4300  
Fax: 0091-(508) 820 4303

## **För prenumerationer**

OMG Publications  
4730 Walnut, Suite 206  
Boulder, Co 80301  
USA

Fax: 0091-(303) 444 3850







# PUBLIKATIONER AV TRIADGRUPPEN

## Verksamhetskrav på informationsadministration

- V 1: IA och verksamhetens krav – erfarenheter från offentlig förvaltning
- V 2: Fallstudie av IA-projektet vid Televerket
- V 3: IA-erfarenheter från företag och myndigheter
- V 4: Den gemensamma informationsmarknaden – en referensram för handlingsfrihet och konkurrenskraft
- V 5: ...fråga är guld. Lokal affärsstyrning utifrån den egna verksamhetens data

## Modellering

- N 1: Modelleringsansatser för begrepps- och datamodellering – Beskrivning och försök till jämförelse
- N 2: Generering av konceptuella modeller från policydokument
- N 3: Espritprojektet Tempora
- N 4: Prövning av regelbaserad metodik inom Posten
- N 5: En kokbok i remodellering – utkast
- N 6: Datorstöd för modellintegration
- N 7: Modellbaserad kunskapsinsamling
- N 8: Modellkvalitet
- N 9: Samband mellan dokument och modeller
- N 10: Modelleringshandboken
  - 1 – Översikt
  - 2 – Modelleringsledarens bashandledning
  - 3 – Modellering i grupp
  - 4 – Kommunikation
  - 5 – Arbetsgångar
  - 6 – Modelleringsväskan
  - 7 – Objektorienterad verksamhetsanalys
  - 8 – Basmodeller
  - 9 – Regelmodellering i praktiken
  - 10 – Business Process Reengineering
  - 11 – Namnsättning
  - 12 – Tolkning av grafiska modeller
- N 11: Ett+Ett=Ett – Två praktikers erfarenheter av modellintegration

## Kunskapsförmedling

- H 1: Handledarutbildning för modelleringsledare, avancerad
- H 2: Slutrapport HUMLA prototyp
- H 3: Utbildning i Informationsadministration
- H 4: Spridning av Hybris – en fallstudie vid Telia

## Uttagssystem

- U 1: Hybris i Unix-miljö
- U 2: DEBRIS
- U 3: Hybris DOS/PimWin på Posten
- U 4: Program för sökning i databaser – en marknadsöversikt
- U 5: Att nå och förstå data – möjligheter och begränsningar

## Katalogprinciper

- K 1: IRDS
- K 2: IRDS Modeller och modellnivåer
- K 3: Koppling begreppsmodell – relationsmodell
- K 4: IBM:s Repository Manager – en Introduktion
- K 5: IBM:s Repository Manager: Datamodelleringsbegreppen
- K 6: IBM:s Repository Manager: Begreppsmodellering i Information Model
- K 7: IBM Repository Manager: Attribut- och värdemodellering i Enterprise Submodel
- K 8: Navigering i Repository
- K 9: TRIAD Newsletter – IRDS inom ISO. Dagsläget
- K 10: TRIAD Newsletter – ISO/IRDS. Händelseutvecklingen 91/92
- K 11: Samverkan mellan resurskataloger – visioner eller behov
- K 12: AD/Cycle I Information Model – Processer och informationsflöden mellan processer
- K 13: AD/Cycle I Information Model – Info Flows inom Processmodellen
- K 14: AD/Cycle I Information Model – Relationsdatabasmodellering
- K 15: AD/Cycle I Information Model – Härlednings-specifikationer i begreppsmodellen
- K 16: IA-prototyp
- K 17: Repository AD/Cycle – International Users Group
- K 18: RAD-konferensen i Chicago, 1992
- K 19: Vad händer inom ANSI-IRDS?
- K 20: Information Warehouse – vad är det?
- K 21: CDIF – en översikt
- K 22: PCTE – en översikt
- K 23: XLII – en öppen och flexibel utvecklingsmiljö
- K 24: Hybris IA/DA – En IA-prototyp vid Telia
- K 25: Introduktion till GDMO-standarderna
- K 26: OpenODB – en introduktion
- K 27: ANSI/X3H7 "Object Information Management"
- K 28: Object Management Group