

Object World West, juli 1997

Stig Berild

Innehåll

1. Konferensen	2
1.1 Allmänt.....	2
1.2 Årets konferens	2
1.3 Inne/ute i år.....	3
2. Inledningsanförande av John Landry, Lotus.....	5
3. Objektorienterad Analys och design (OAD).....	7
4. CORBA	8
5. Komponenter	9
6. Objektorientering ingen ”silver bullet”	11
7. Övrigt av intresse	12
8. Sammanfattning	13

1. Konferensen

1.1 Allmänt

Konferensen Object World startade som en årlig konferens under tidigt 90-tal med placering i San Francisco. Arrangören Object Management Group (OMG) såg ursprungligen konferensen som en marknadsföringskanal. OMGs primära strävan är ju att åstadkomma standarder för systemsamverkan mellan komponenter (objekt) i en distribuerad och heterogen miljö baserat på ett objektorienterat synsätt. Samverkansaspekten har också förts fram som en central ingrediens i konferensprogrammet. Skickligt nog valde man dock att etablera Object World på den då intensivt framforsande objektorienteringvågen genom att formulera ett mycket brett tekniktema. Benämningen med "Object" i centrum var knappast slumpmässigt valt. Inriktningen var också påtagligt industriellt inriktat med fokus på existerande lösningar och produkter, vunna erfarenheter, formulerade standarder. Tonvikt lades mer på state-of-the-art-föredrag med dokumentation i form av ljusbilder än på forskningsrapporter. På så vis uppstod ingen allvarlig konkurrens med den andra stora konferensen med ett objektorienterat (OO) tema, nämligen OOPSLA (Object Oriented Programming Systems, Languages and Applications) eftersom OOPSLA huvudsakligen lockar akademiker med programmeringsspråksintresse. Framgången lät heller inte vänta på sig. Efter ett par år dubblerades konferensen i Boston. Därefter expanderade man ytterligare genom att låta konferensen gå på "turné" till ett antal platser i världen, alla under ett och samma kalenderår. Förra året tog så den professionella konferensarrangören SOFTBANK COMDEX över driften från OMG. OMG spelar dock fortfarande en mycket aktiv roll som sponsor och i aktiv medverkan.

1.2 Årets konferens

Över till årets konferens. Traditionsenligt är San Francisco-konferensen den största av årets Object World-konferenser. Ca 8000 deltagare totalt inklusive utställningsbesökare räknades in. Utställningen omfattade ett sextiotal montrar, från de stora aktörernas flashiga budskap och världsbästa produkter till de små fåmansföretagens konsulttjänster.

Detta är konferensen för dem som på något sätt kommer i kontakt med objektorienterade produkter, metoder, modeller, et c i samband med någon verksamhetsrelaterad aktivitet. Akademiker och "finlirare" söker sig till OOPSLA. Proper klädsel går före jeans och sandaler.

Överlag kändes konferensen något avslagen. Objekt är inte längre intressanta i sig utan vilka problem de kan lösa. Kanske har konferensen blivit profillös nu när objektorientering genomsyrar det mesta inom IT, ex programmeringsspråk, databaser, modellering, användargränssnitt, distribuerade system.

Parallellt med Object World pågick i angränsande lokaler Informixs användarförenings årssammankomst med betydligt större utställningsyta (dubbelt så många utställare). Dessutom med inbjudna talare som General Colin Powell. Det är produkter och lösningar man söker idag, inte akademiska visioner och komplicerat finlir. OO-språket Java genererar som konferenstema

ofantliga deltagarskaror. Inte enbart för att det är ett nytt spännande programmeringsspråk att göra kul saker med (även om detta säkerligen varit en framträdande drivkraft till en början), utan för att språket representerar helt nya förutsättningar och nya möjligheter att realisera existerande verksamhetsbehov av IT-stöd.

Som intressant statistik kan noteras att konferensen ”Java Internet Business Expo” med stort fokus kring Java samlade ca 15000 deltagare i augusti 1997. Samtidigt drar det sedan många år etablerade årliga OOPSLA endast ca 2-3000 deltagare. Konferenser måste numera ha ett konkret budskap samt redovisa angelägna lösningar - en allmän temarubrik lockar inte längre, i alla fall inte under ett större antal år. Kan konferensen sedan rida på en massmedial attraktionsväg är lyckan gjord.

Object World har de senaste två åren i ökande utsträckning försökt anpassa sig till rådande trender. Exempelvis finns en dragning från tekniska infrastrukturlösningars egenvärde oavsett om de heter CORBA, DCOM, DCE eller Tuxedo till ett nymornat intresse för att placera in infrastrukturen som nyttgörare i ett verksamhetsperspektiv. ”Business Object” är exv ett populärt begrepp. Dels ligger orsaken i att de primära infrastrukturproblemen nu lösts, dels i ett sunt konstaterande att infrastruktur inte är något självändamål i sig - den ska fylla ett behov, vara till nytta – som middleware - för verksamheten. Alldeles följdriktigt kom årets tema att formuleras ”Distributed Applications for the Enterprise”.

Ännu mer påtaglig blir denna nyorientering nästa år då SOFTBANK COMDEX har valt att placera in Object World West som en del av en betydligt större och bredare satsning under rubriken ”COMDEX/Enterprise”. Den långsiktiga trenden mot verksamhetsrelaterade, distribuerade systemmiljöer är alldeles klar.

1.3 Inne/ute i år

- * Komponenter i största allmänhet, speciellt komponenter i Internet-miljö är hett stoff. Se vidare avsnitt 5.
- * ”Business centric view” har tagit över den tidigare teknikfokuseringen.
- * Intranet/extranet är självklara begrepp för den invigde. Att dessutom föra in begreppen supernet och meganet i resonemangen visar på trendnäsa. Dessa begrepp har ännu ingen definition men låter avancerade. De refererar ju på något sätt till nästnästa generation infrastrukturer. Den som även lyckas föra ett kortare resonemang om vikten av network management i denna miljö lyser av professionalism.
- * Bäst visar man sina gedigna kunskaper inom OMGs verksamhet och de specifikationer som där tas fram genom att prata om IIOP (Internet Inter-ORB Protocol) och dess banbrytande roll för de framtida globala tillämpningsmiljöerna. Ett litet exempel på en intressant objektmiljö formulerad i IDL (Interface Definition Language) skadar inte heller.

- * Apropå IIOP och CORBA; Object Request Brokers (ORBs) går under epitetet middleware. Liksom så mycket annat. Det är fint att vara middleware-produkt. Bland andra typer av middleware återfinns Database Connectivity middleware, TP Monitor middleware, Messaging middleware, Remote Procedure Call (RPC) middleware och Publish and Subscribe middleware – var och en med sina specifika egenskaper. Trenden mot allt rikhaltigare funktionalitet inom varje kategori suddar dock alltmer ut distinktionerna.
- * ”Legacy system integration” tillhör numer de seriösa diskussionspunkterna. För inte länge sedan var ”legacy system” beteckningen på något fult: en gammal, otymplig, tung tillämpning, en barlast som snarast möjligt borde skrotas. Detta gällde tills det blev rumsrent att konstatera att dessa tillämpningar visserligen var gamla, oöverblickbara och tunga, men de utförde sitt arbete med den äran och har så gjort under en följd av år. Att de sedan ofta representerar den mest affärskritiska delen av en verksamhets IT-stöd missgynnar knappast dess rykte. Till råga på allt är de så komplexa och innehåller så mycket av verksamhetens vitala affärsregler att en omskrivning i modernare tappning knappast låter sig göras utan stora uppoffringar. Re-engineering är inte att tala om. Alltså bör legacy systems hanteras varsamt. De bör med fingertoppskänsla föras in i och integreras med den modernare systemmiljön, knappast raseras – i alla fall inte nu och framförallt inte på en gång.
- * För övrigt kan man med fördel använda såväl ”Enterprise” som ”Universal” som prefix till de allra flesta ord, exv Universal Server, Enterprise View, Enterprise User, Universal infrastructure,
- * Objekt och objektmodeller är i största allmänhet uttjatade begrepp. Inte för att de lyser av färdigpenetrerad, väldefinierad kraft - snarare har man accepterat att varje behov har valt sin version av betydelse och innehåll. Samtidigt lämnas stringens och entydighet till den fortfarande intresserade att sopa ihop och i möjligaste mån skapa något vettigt av. På något sätt kännetecknas OO-området av otålighet, starka behov av lösningar inom många olika tillämpningsområden. Dessa tillåter sig inte invänta de mest välgrundade och teoretiskt förankrade plattformarna. Snarare fångar man i rusningen mot framtiden upp något rimligt användbart, lockad av OO-modellens förledande enkla, attraktiva budskap.

I ärlighetens namn finns dock åtminstone ett stort undantag. Det är Unified Modeling Language, en begreppsmodell som omfattar det mesta som i begreppsväg behövs för att formulera system utvecklade enligt objektorienterad analys och design (se vidare avsnitt 3).

- * OO-språken har knappast haft någon framträdande plats inom Object World. Möjligtvis C++ vars breda användning fört fram det som ett de facto standardspråk. Även Smalltalk har rönt visst intresse för dess elegans och enkelhet. Detta intresse håller nu raskt på att svalna till en plats bland Eiffel, ADA, Objective C, m fl. som konferensattraktion i takt med Javas snabbt stigande stjärna och infrastrukturens språkoberoende samverkansmöjligheter.
- * Java förresten; Java refereras så till den grad att tröttnad inträder. Java är långt ifrån uttjatad, men det kan bli tungt i för täta doser. Konferensdeltagarna tycks överlag synnerligen medvetna om att det finns en stor, ja än så länge våldsamt mycket större, så kallad konventionell IT-värld att beakta både för dess relevans och för dess kommersiella bärkraft. Object Worlds

konferensdeltagare önskar se och kanske inhandla lösningar på existerande problem. I deras ögon är det primära att bevisa vad Java kan vara bra för i tunga tillämpningsmiljöer. Java-plattformens alla spännande faciliteter representerar i det perspektivet knappast ett tillräckligt egenvärde.

2. Inledningsanförande av John Landry, Lotus

Konferensen inleddes med ett anförande av John Landry från Lotus under titeln ”Embracing New Frontiers in Systems and Application Development”. Pompöst, javisst. Å andra sidan ska ett inledningsanförande vara visionärt – skapa kittlingar i maggropen och en känsla av upprymdhet över att just denna konferens är den breda vägen in i någonting stort och betydelsefullt. Och över att just du är en lyckans ost som valt att komma hit för att bevittna och ta del.

Den mycket sympatiska och välartikulerade Landry valde att beröra ett antal aktuella trender enligt följande:

Alla talar om den förestående kampen mellan vanlig PC och NC (Network Computer) i de framtida hemmen. Givetvis bör denna diskussion föras även på Object World i anslutning till den inte speciellt avlägsna visionen om helt mobila system(-komponenter) och likaledes mobil information över ett globalt nätverk och sammanbundet, kontrollerat i en CORBA-miljö.

Landry ville se en tredje kategori på spelplanen, nämligen Special Purpose NC's. Dessa tänks omfatta mer specifika egenskaper framsprungna ur konkreta behov och rörelsemönster till skillnad från den genererella funktionalitet PC och NC erbjuder. Därmed kan användargränssnittet anpassas och förenklas, något som vanliga användare attraheras av. Det finns många behov där en PC eller NC är att skjuta ”mygg med kanoner”. Även om en NC syftar till enkelt handhavande i sig är därmed bara halva vinsten nådd. Den andra halvan beror ju på de tillämpningar som i praktiken körs. Eftersom en NC i realiteten ska kunna hantera i stort sett samma tillämpningsmiljöer som en PC blir skillnaden i detta avseende marginell. Som exempel på en Special Purpose NC nämnde Landry en integrerad epost/fax/telefon/PDA, specialdesignade informationsinsamlingsutrustningar, m m.

Värt att betänka är också den säkerligen lavinartade ökningen av datorer i all slags utrustning från hushållsapparater, hemelektronik, låsmekanismer, bilar, garageportar, telefoner, som alla kommunicerar på en CORBA-plattform, gärna med funktionaliteten representerad i Java applets.

I enlighet med konferensens verksamhetsorienterade anda anslog Landry ett antal infallsvinklar kring elektronisk handel. De är i sig knappast sensationella men speglar ändå den klara trend som nu kan skönjas bort från infrastrukturens alla interna intrikata sublimiteter till dess användning för fyllande av alla upptänkliga verksamheters mångskiftande behov.

En konsekvens av ITs intåg i varje hörn av samhället är exv det minskade behovet av mellanhänder. Landry trodde på en snabb förändring i kundattityder och servicemönster. De som idag växer upp upplever IT som något positivt och har därför förmågan att ta till sig ny teknologi, dess användning och dess kanske delvis omvälvande konsekvenser för det vardagliga livet. Leverantörer kommer i

betydligt större utsträckning att kunna ta direktkontakt med kunden via elektroniska infrastrukturer. I den mån mellanhanden kan erbjuda något mervärde kommer den att ha ett affärsberättigande. Detta har också poängterats som en viktig faktor som dämpare av alltför vidlyftiga visioner. Många människor föredrar fortfarande kontakter med och råd från människor istället för från en artificiell miljö. Människor har vi vant oss att kommunicera med från födseln, dock knappast datorgränssnitt (än så länge?!?!).

Å andra sidan kan det mycket väl komma att bli så i vissa verksamheter att service, vägledning och informationsstöd kan erbjudas med större kvalitet elektroniskt än i konventionella miljöer. I anslutning till en bok av intresse kan den presumtive kunden enkelt finna fakta om författaren, referenser till andra alster, val mellan text- eller talbok mellan leverans elektroniskt eller i tryckt form, recensioner, upplagor, vilka språk den erbjuds på, osv - service som åtminstone delvis kan vara ganska svår att få i en bokhandel. Begreppet "Knowledge Based Purchase" börjar användas.

Bland exempel på försvinnande mellanhänder eller åtminstone sådana med betydligt beskärda uppgifter, nämner Landry olika typer av mäklare (fastighets-, aktie-, ...), resebyråer (exv <http://expedia.msn.com> eller flera av de större flygbolagens hemsidor), bokhandlare (exv <http://amazon.com>), bankkontor (exv <https://banking.wellsfargo.com/>). Wells Fargo var först med en helt CORBA-baserad lösning), musik/videoaffärer. Redan idag bedrivs en snabbt stegrande handel inom dessa kategorier.

En snarlik trend gäller övergången från "product to service". Ett påtagligt exempel är fotoframkallning. Istället för konventionell leverans av korten kan alternativt leverans ske elektroniskt i digital form till önskad epost-adress. Kanske räcker det med leverans av en unik identifikation av de hos framkallaren förarade bilderna. Servicen kan bestå av möjligheter att plocka ut endast de bilder man gillar, flytta över dem till ett eget lokalt eller hos framkallaren abonnerat "familjealbum" (mer eller mindre strukturerat), möjligheter att klippa och klistra i bilderna, vidaresändning till svärmor, med mycket mera. Det vi är villiga att betala för är servicen snarare än ett kort på papper. För övrigt är det inte osannolikt att även framkallaren blir en bortrationaliserad mellanhand när den digitala kameran är var mans egendom och servicen kan erhållas med hjälp av ett till den egna datorn nerladdat applet.

Ytterligare en trend gäller övergången från leverantörstyrd till konsumentstyrd handel. Som exempel nämndes bilaffärer där köparen elektroniskt kan ta fram all multimediebaserad information om intressanta bilkategorier med hjälp av mer eller mindre avancerad sök- och urvalsmekanismer. När önskad biltyp valts läggs en offertförfrågan ut på nätet, leverantörerna offererar, varefter bästa erbjudande väljs. Den önskade provkörningen tillhandahålls genom en ny affärsidé för provkörningsuthyrning. Försäkringar, vin, hotellvistelser, m m kan upphandlas på liknande sätt. Se exv www.carpaint.msn.com.

Kanske kommer helt andra mekanismer att etableras. Vad som än blir resultatet kommer det att innebära en revolution med helt nya förutsättningar för rollen som producent, konsument, mellanhand, ... Och det kommer att ske snabbt. Vissa bedömare - säkert i och för sig optimistiska (Oracle?), men ändå - anser att vi kommer att ha ca 500 miljarder NC kring år 2000. Kom också ihåg att mycket stora grupper människor i världen inte längre är datoromogna. De kan PC, video, avancerad telefoni, De kommer snabbt att absorbera ny teknologi och utnyttja ny service och

tjänster. Se på mobiltelefonutvecklingen. Å andra sidan kommer den mycket stora grupp, förmodligen köpstarka, konsumenter som ryggar för tekniken, som ser allt utom av/på-knappen på videon som oöverstigligt hinder, att borga för en kontinuitet och lugn övergång för dem som så önskar. Inte sällan har det ju också visat sig att IT-folket låtit sin fantasi och teknologi rusa i väg utan hänsyn till de realiteter som styr användarbeteenden. Framtiden får utvisa.

3. Objektorienterad Analys och design (OAD)

Självfallet fanns temat OAD också med på ett hörn. Temat är för närvarande höggradigt aktuellt i och med OMGs ansträngningar för att etablera en standardiserad begreppsmodell för OAD. Begreppsmodellen går under beteckningen UML (Unified Modeling Language). UML är en specifikation av de modelleringsbegrepp som bedöms komma till användning för att formulera olika modeller inom olika faser av en objektorienterad systemutvecklingsaktivitet. Tanken är att begreppen ska vara så heltäckande att oavsett metodansats samtliga erforderliga begrepp ska kunna återfinnas som en delmängd av UML. Ett antal företag (Rational, IBM, Platinum, Microsoft, H-P, m fl med Rational som huvudaktör) står bakom förslaget. Med största sannolikhet kommer det att antas av OMG före årsskiftet. Vid OMGs septembermöte i Dublin bestämde Domain Technical Committee att genomföra en slutlig omröstning bland samtliga 96 röstberättigade (för en acceptans av den version som betecknas UML 1.1). Ja-rösterna måste vara dubbelt så många som nej-rösterna för att förslaget ska lämnas vidare för slutligt avgörande av OMGs Board of Directors (BoD). När detta skrivs har 36 ja- och inga nej-röster avgivits. BoDs formella godkännande i början av december bedöms i detta fall bli en formalitet. UML 1.1 har föregåtts av ett intensivt kompromissarbete under våren och sommaren mellan de företag som står bakom förslaget. Ursprungligen inlämnades sex separata förslag från olika företag(skonstellationer) in till OMG. Att dessa kunnat sammanjämkas till en enda slutlig specifikation vittnar om en effektiv process och en stark kompromissvilja. Hela metodområdet för OAD stod annars inför ett sammanbrott vad gäller ensningsmöjligheter men framförallt vad gäller dess allmänna trovärdighet.

En sammanjämkning av alla mer eller mindre överlappande begrepp från olika metodansatser välkomnas säkerligen av alla. Leverantörer av produkter önskar det för att området som helhet får en seriösare status och därmed ökad attraktionskraft på kunderna. Metodprofeter önskar det därför att de sannolikt är högeligen trötta på all debatt som aldrig leder någonstans. Användare och utvecklare önskar det för den ökande begreppsstabiliteten och för ökat produkt/metod-oberoende (i alla fall i teorin) samt enklare möjligheter till modellutbyte mellan verktyg och snabbare förståelse mellan användare från olika utvecklingsmiljöer. Samma begreppsapparat tillämpas ju förhoppningsvis även om de interna mekanismerna, användargränssnittet, m m skiljer sig åt mellan produkter.

Kompromissarbetet har lämnat efter sig vissa luckor som under lugnare omständigheter behöver kompletteras. Kompletteringsarbetet kommer att utföras inom den grupp som aktivt medverkat vid framtagningen av UML 1.1. Sannolikt kommer resultatet att betecknas UML 1.2.

Alla är inte helt entusiastiska över UML. UML innehåller ett stort antal mer eller mindre relaterade begrepp. Svårt att hitta, förstå, risk för missförstånd. Detta löses delvis genom att skapa subgrupper av begrepp för olika ändamål med vidhängande pedagogiska förklaringar. Ett annat problem gäller

målsättningen. Genom att försöka vara heltäckande istället för att representera en kärna kommer kontinuerliga behov av kompletteringar och revideringar att krävas. OMGs princip i andra sammanhang bygger på en fastslagen grundmodell som alla specifikationer har att svara mot samt påbyggnader (profiles) för varje enskilt behov. Kunde kanske samma principer ha applicerats här? Morgondagens tillämpningar och metoder kommer knappast att överensstämma med dagens. Ingen kan förutse denna framtid. Kanske ökar en omfattande begreppsmodell risken för att en standard låser fast en annars vital vidareutveckling av områdets begrepp enligt olika profiles. Metoder och begrepp måste tillåtas utvecklas som en följd av användningserfarenheter m m. Kanske kommer en ökad komponentbaserad utveckling, användning av ramverk, distribution, replikering, m m att erbjuda helt nya förutsättningar och därmed nya uppsättningar begrepp – för att inte tala om effekterna på utvecklingsmetoder.

Nåväl, UML har fyllt ett begreppssamordningsbehov. Därutöver fyller UML en alldeles naturlig roll som ”neutral begreppsinstans” i samband med överföring av modeller mellan olika Case-verktyg enligt samma syfte som den mångåriga CDIF-standarderna. CDIF lyckades dock aldrig formulera en lika fullödig begreppsmodell för OAD.

Någon myntade uttrycket ”UML is like a Swiss Army Knife - doing a lot of things but nothing really well”. Kanske ligger det något i detta. Å andra sidan är det alltid lättare att kritisera än att vara konstruktiv. Alternativet med en vild flora ”frisvägande” ansatser och begrepp vill knappast någon ha tillbaka.

4. CORBA

OMGs konkreta förslag kring distribuerad objektsamverkan har formulerats i två specifikationer CORBA 1 och CORBA 2 (CORBA= Common Object Request Broker Architecture). CORBA 1 som godkändes redan 1991 svarade upp mot samverkan inom ett nät. CORBA 2 kom 1994. Det är ett komplement innehållande generellt protokoll mellan nät (GIOP) samt ett specifikt Internet-protokoll baserat på Internets eget TCP/IP. Det senare går i folkmun under IIOP (Internet Inter ORB Protocol). Därutöver tillkommer ett antal Object Services och Common Facilities. Se vidare SISU-rapport 97:08 ”OMG – vad, när, hur?”.

Uppskattningsvis finns ca 20-30 ORB-produkter med egenskaper enligt CORBA 1. Flertalet uppges också klara CORBA 2. De har i olika omfattning orkat realisera specificerade Services och Facilities. Produkterna har tidigare bedömts vara instabila och/eller prestandasvaga. De har genomsyrats av en kontinuerlig utveckling i jakten på att i möjligaste mån svara upp mot OMGs i strid ström utgivna specifikationer. Leverantörerna har ofta varit små, resurssvaga och med osäker överlevnadspotential. Alla bedömningar pekar på att en utslagning kommer att ske. Naturligtvis skapar detta en avvaktande hållning från presumtiva kunders sida. Alternativt närmar de sig Microsofts kanske inte lika eleganta men desto säkrare motsvarande lösningar. Man väljer bort det öppna havet mot insjöns relativa lugn även om där inte finns några farbara utlopp. Kanske hoppas man på att en kanal mellan insjön och havet så småningom grävs.

Av flera skäl har nu vinden börjat vända. CORBA-leverantörerna utstrålar ett påtagligt stort självförtroende – i alla händelser utåt. Såväl teknologi som produkter har kunnat bevisa sin

ändamålsenlighet och stabilitet i ett antal stora, kritiska tillämpningar. En banderoll konstaterade att "Distributed Object Computing is maturing surprisingly fast".

De stora IT-aktörerna som SUN, IBM och Digital (vars ObjectBroker sedermera köpts av BEA Systems) har bundit upp sig för strategiskt avgörande satsningar baserade på CORBA. Detsamma gäller Netscape som licensierat Visigenics JAVA-baserade CORBA-klient för inkludering i samtliga bläddrare. Glöm heller inte bort att Microsoft genom sitt motsvarande DCOM bidrar till att etablera principerna om distribuerade samverkande systemkomponenter som den givna framtida vägen att välja. Riktningen framåt är alltså klar. Samtliga aktörer drar åt samma håll. Om kunden sedan väljer en DCOM- eller CORBA-lösning är i det perspektivet av underordnad betydelse och underställt helt andra kraftfält. Många anser för övrigt att det är en tidsfråga innan ORBs som middleware kommer att vara inbyggda som automatisk service i andra typer av mer verksamhetsnära produkter. Därmed kommer de från ett tillämpningsperspektiv att både vara osynliga och sakna egen fascinationskraft.

Kunderna sägs numer våga lita på CORBA-teknologins bärkraft och påstås strömma till. Ett antal lyckade realiseringar finns erfarenhetsbelagda. Wells Fargo Banks kundstödssystem är ett exempel. Se vidare www.wellsfargo.com och www.cushing.com.

Middleware är för närvarande under häftig turbulens. Teknologin är synnerligen komplicerad. Ett antal mer eller mindre sublima kontaktytor mot omgivande teknologier underlättar knappast utvärderingsläget. Förutom den tekniska bedömningen tillkommer att göra en kommersiell prognos med sikten mot framtiden. Prognosinstituten famlar på målet. Ena året höjer exv Ovum CORBA till skyarna. Nästa år är det knappt värt vatten, knappt heller DCOM. Nu plötsligt lyser stjärnan över två ännu oprövade lösningar.

En presumtiv kund rekommenderas dels att ta del av allt som skrivs med "filterglasögonen" på, dels att utföra reella tester med hänsyn till egna förutsättningar. Endast nästan helt kompatibla ORBar från olika leverantörer kan innebära katastrof. Att lita på att nästa generation av en ORB-produkt ska innehålla lösningen på alla nuvarande problem är likaså att ta risker. Å andra sidan är höggradig diversifiering, i alla händelser i större verksamheter, en realitet som kräver öppna lösningar. Att tro att läget kommer att vara mer ensat i framtiden är att tro att historien inte upprepar sig.

Vi får heller inte glömma att teknologin plötsligt och utan egentlig möjlighet till förvarning kan ta nya banor. Den allt starkare Internet/CORBA/Java-integreringen kan exv visa sig skapa helt nya, spännande förutsättningar för globala tillämpningsmiljöer. Ansträngningar för DCOM-CORBA-samverkan pågår inom OMG. Finns den riktiga viljan från alla parter sida, finns lösningarna.

5. Komponenter

"Component is the goal as in all other industries." "Components are cool." konstaterades frejdigt vid en paneldebatt. Att komponenter vuxit fram som en tilltrodd förlösare på den planhalva objektorienteringen mutar in men inte lyckats klara av är alldeles påtagligt. Objekt och

objektmodellen har tyvärr i huvudsak fått träda fram som en teknikorierad struktureringsfacilitet, knappast som en modell för verksamhetsorienterad systemkonstruktion. Verksamhetens företrädare ställde förhoppningar till en enkel, kraftfull och synnerligen påtaglig modell som de dessutom, i alla händelser i tanken, enkelt kunde applicera på "sin värld". En förhoppning som inte infriats. Till stor del har de objektorierade programmeringsspråkens företrädare sig själva att skylla. Allting, från det minsta lilla heltal till den största tillämpning har schablonmässigt klassats som objekt. Vem finner tanke- och struktureringsstöd inom denna totala skala? Det verksamhetsorienterade företrädare såg som idén bakom objektmodellen formulerar man nu istället som en förhoppning gentemot komponentansatsen.

Komponentmodellen tycks sakna något av objektmodellens uttrycks kraft, exv dess generaliseringsdefinitioner. Kanske behöver man inte alltid klass(typ)-begreppet. Varje komponent kan vid behov vara en förekomst i sig (dvs mer "object based" än "object oriented" enligt Wegners gamla taxonomi). Komponenter är också betydligt mer självständiga och ickeberoende företeelser än objekt, gärna höggradigt distribuerade. Den borte och inte helt unge teknikern skönjer nu klara likheter med tidigare världars så flitigt använda subrutiner. Jomennisvis säger man, men ändå inte. En komponent är nämligen mer verksamhetsorienterad och -motiverad än subrutinen, vars historia har varit för system/funktionsfixerad. Däri ligger säkert en poäng. Bekymret för den stackars åhöraren är att ett nytt begrepp som "komponent" snabbt blir upptaget av församlingen som representerande något gott, användbart och bra utan att någon på vägen försökt sig på en mer preciserad definition. "Kanske" inflikades ovan, just på grund av att ingen entydig definition av "komponent" finns – mig veterligt. På ett övergripande plan är det lätt att bli överens. Komponenter är bra för de delar upp något mer komplicerat i ett antal mindre komplicerade beståndsdelar. Därutöver genererar alla sin uppfattning om innebörden. Efter mycket kort tid har begreppet satt sig - alla slänger sig med det - och då är det plötsligt för sent att fråga vad begreppet innebär utan att väcka löje.

Precis så är läget för "komponent". Objekt räcker inte till, klarar inte sitt syfte, alternativt har ett annat syfte. Vi måste ha något mer rejält, substantiellt, m a o komponenter precis som i annan industri. Fortfarande tycks spelet i huvudsak avgöras på teknikernas spelplan. De något mer verksamhetsinriktade har istället tagit till sig begreppet affärsobjekt (Business Object). Kanske är "komponent" och "affärsobjekt" bara olika sidor av samma mynt, kanske är de ordentligt olika företeelser. Det går aldrig att få reda på eftersom de båda planhalvorna nästan aldrig möts, i alla fall inte med någon hjärtligare vilja.

Grovt sett tycks man mena att en komponent står för en verksamhetsrelaterad, från såväl användarsom utvecklarsidan gripbar funktionalitet. Med verksamhetsrelateringen följer att komponenten normalt är en betydligt mer sammansatt och större företeelse än ett objekt som ofta består av relativt få kodrader. Detta behöver dock inte innebära ett komplicerat komponentgränssnitt. Snarare bör strävan vara den motsatta. Den viktiga skillnaden ligger dock i syftet. Objekten är systemkomponenter i syfte att vara en kugge i en systemfunktionalitet medan komponenten representerar en direkt verksamhetsrelaterad företeelse eller service.

Vi kommer sannolikt att se en framväxt av olika typer av ramverk för olika tillämpningsområden alternativt standardsystem som är anpassningsbara. Med dessa bortfaller behovet av ett renodlat objektorierat angreppssätt annat än för att lösa komponentens egen interna funktionalitet.

Ramverket erbjuder mycket enkla inpluggningsmöjligheter av erforderliga kompletterande komponenter. Komplexiteten, löst genom en objektorienterad eller annan ansats ligger i ramverkets fördolda.

Uttestning av tillämpningar uppbyggda av en uppsättning komponenter kommer att bli både enklare och svårare än konventionellt utvecklade tillämpningar. Enklare därför att komponenter som köps eller återanvänds har testats och därför sannolikt motsvarar rimliga kvalitetskrav. Enklare eftersom även övriga komponenter är testbara utifrån ett entydigt gränssnitt. Det svåra ligger i utvärderingen av komponenternas samspel. Komponenternas olika ursprung (köpt, kopierat, återanvänt, ...) bygger på bekymren. Vem ansvarar och hur? Hur sker versionshantering av respektive komponent och hur anpassas denna till den egna tillämpningen? Får, och i så fall hur får komponenter anpassas/vidareutvecklas för svara upp mot specifika tillämpningskrav? Får och i så fall när och under vilka villkor får flera tillämpningar utnyttja samma komponent? Osv. Dessutom har vi att beakta alla de effekter som komponenters plug-and-play-egenskaper uppmuntrat till i form av snabb och flexibel anpassning till kontinuerligt nya förutsättningar. Tillämpning som begrepp formar en allt otydligare skepnad.

6. Objektorientering ingen "silver bullet"

Objektorientering, OO-språk mm, hjälper knappast - snarare tvärtom - om intresset för seriös analys och design är ringa eller helt saknas i anslutning till utveckling av tillämpningar. Inte ovanligt bland framförallt programmerarskrået tycks vara uppfattningen att med ett OO språk blir det så lätt att formulera lösningen och en så elegant struktur mer eller mindre automatiskt att de tidiga faserna blir onödiga. För övrigt tycks fortfarande gälla att OAD upplevs som flummigt tjafs. Allt enligt Norman Kerth, en av föredragshållarna.

Enligt Kerth styrs fortfarande både små och stora tillämpningsprojekt av tekniker med höggradig misstänksamhet mot de tidiga faserna. Resultatet blir följdriktigt fortfarande att 80% av de totala resurserna för en OO-baserad tillämpning går åt för underhåll av dåliga system. Lite tillspetsat gäller fortfarande det upplysta året 1997 att kod regelmässigt skrivs först varefter man vid första förändringstryck gör reengineering eller annat försök att förstå vad man åstadkommit. I en hel del fall har en OAD-fas genomlöpts dock utan att dess resultat fått ligga till grund för kodningen. Vattentäta skott gäller ofta mellan dem oavsett hur vackert de objektorienterade systemutvecklingsmetoderna beskriver modellkontinuitet över systemets livscykel.

En förutsättning för ett bra system oavsett alla tekniska och metodmässiga förutsättningar är att det byggs med en kunskap om och förståelse för verksamheten och dess behov.

En ökad orientering mot komponentuppbyggda system kommer att kräva avsevärda kompletteringar/revideringar av existerande systemutvecklingsmetoder. Sannolikt kommer dock komponentbaseringen att resultera i betydligt mer användarvänliga och flexibla system. Detta beroende på att komponentmodellen är betydligt mer gripbar och enklare att ta till sig än objektmodellen.

Blir äntligen återanvändning en realitet?

7. Övrigt av intresse

- * OMGs chef Chris Stone lämnar OMG för att börja på Novell. Efter ett mycket framgångsrikt pionjärarbete som ledare för OMG under alla dess år fram till nu har Stone förlänats titeln "Founder". En del anar en osäkrare framtid för OMG. Andra ser en helt normal utveckling. Chefer bör inte sitta i oändlighet. Dessutom är det en sak att starta och bygga upp en organisation, något helt annat att föra en nu mycket stor organisation framåt mot existerande och nya mål med både kontinuitet och framsynhet. OMGs nye VD, Richard Soley, fyller med stor auktoritet den rollen. Soley har direkt under Stone varit den reellt ansvarige under alla år för de arbeten som bedrivits med sådan framgång. Soley har dessutom efterhand kommit att spela en alltmer dominerande roll inom OMG. Skulle OMG börja vittra sönder eller mista sin kraft är det knappast på grund av det just genomförda VD-bytet. Riskerna kan snarare hämtas från dess storlek med idag över 750 betalande medlemmar samt alla parallella aktiviteter de olika undergrupperna (Task Forces) idag engagerar sig i. Den starka inbrytningen på Business Object-området innebär en ny utmaning. Här närmar man sig verksamhetsrelaterade teman som kanske inte lika enkelt låter sig standardiseras som generella infrastrukturlösningar. Här finns heller ingen tradition och kontinuitet att falla tillbaka på. Snarare träder man in på andra organisationers revir och kompetens. Hur väl är man förankrad inom respektive affärsområde? Gäller den förankringen internationellt? Svaret lär växa fram under de närmaste två till tre åren.
- * Återanvändning är ständigt på ropet. Förvånansvärt lite av lyckad återanvändning finns att redovisa. En förutsättning för formulering av reellt återanvändbara objekt anses vara gedigen erfarenhet från objektens användning i ett antal lyckade systemutvecklingsprojekt under vilka objekten successivt givits allt fastare form. Annars blir det bara gissningar, en lyx ingen har råd med. Flera påpekade att återanvändningsansvar bör tillställas en separat organisation med just detta som mål. Den måste sälja dessa objekt aktivt - "run like a business". Vanliga projekt har varken tid, incitament eller behovsöversikt annat än möjligtvis inom eget projekt. Andra menar att reell återanvändning bara bör eftersträvas och i realiteten komma till stånd för verksamhetsrelaterade komponenter - affärsobjekt - utom i mycket specifika fall som exv klassbibliotek för grafik. Affärsobjekten har en verksamhetsförankring och -relatering som gör dem till naturligt stabila byggstenar.
- * Varje OO-konferens talar vidlyftigt om alla "paradigm shifts"; om den svåra men kloka vägen in i objektorienteringens mjuka famn, en väg som kräver mycken stöd och klokskap; om behovet av styrning, ledning av OO-projekt; om alla fiender som lurar bakom hörnen som farliga bakåtsträvare, o s v. Ok, varje övergång till något nytt är en komplicerad process. Men kan inte oron också vara en indikation på att OO inte är denna sagolika silver bullet förhoppningarna ställ i sikte. Visserligen håller säkerligen gamla system och gamla förfaringssätt knappast måttet men är det så säkert att det nya är så mycket bättre? Kanske löses här vissa problem medan andra nygenereras. Det är enkelt att kritisera "legacy systems" som något gammalt, oformligt, dåligt - d v s något man med alla medel vill bort ifrån. Att OO kanske skapar andra typer av problem men med samma dignitet förs aldrig på tal eftersom det är en reflexion som knappast ligger den borne OO-entusiasterna speciellt nära.

Faktum är ju att många av dessa legacy monster fungerar som den livsnerv hela verksamheters överlevnad baseras på. Dessutom fungerar de inte sällan bra för sitt ändamål samtidigt som framförallt stora OO-baserade projekt redovisar problem av olika slag. Säkerligen innebär heller inte OO det slutliga paradigmskiftet. Att även den nya fina OO-tillämpningen snart träder in som kandidat till epitetet legacy system borde därför inte upplevas som ett främmande perspektiv.

I samma OO-framåtanda talar man om de ”gamla” systemutvecklarna som lika legacy som de system de utvecklat. I vissa lyckliga fall kan de omskolas, i normalfallet avyttras på de förbrukades och oanvändbaras avfallshög. Att deras tvekan kan vara erfarenhetsbaserad och botten i kloka bedömningar om OO's bräcklighet i mer komplicerade systemmiljöer - en kompetens att ta tillvara - är en avlägsen tanke.

- * Den nya frihet som Internet, IIOP, JAVA m m ger realiserare av distribuerade tillämpningar och de nya problem denna frihet för med sig kommer att kräva designstöd, realiseringsstöd, managementstöd, m m av en helt ny dignitet. Case-verktyg, utvecklingsmetoder har inte hängt med i den snabba utvecklingen. Snabb förnyelse är att vänta.
- * Alla talar om informationssamhället och om informationen som den verkligt vitala resursen i en verksamhet. Med treskiktssarkitekturens uppdelning i användargränssnitt – affärslogik/regler – databas poängteras alltmer vikten av att skapa, kontrollera och underhålla mellanskiktet, dvs affärsreglerna och då gärna packeterade som affärsobjekt. Det kommer nämligen att finnas två typer av tillämpningar; dels allmängods, dels den differentierade, verksamhetsavgörande funktionaliteten. Vitala affärsobjekt kommer att mejslas ut med yttersta noggrannhet och under rigorös kontroll. Därutöver kommer en mängd mindre kritiska och/eller temporära affärsobjekt att kreeras för olika behov. Utvecklingen av denna senare typ bedöms bli explosionsartad.

8. Sammanfattning

All diskussion om den nödvändiga verksamhetsorienteringen till trots, präglades konferensens anda lite tillspetsat av dess deltagare, dvs av teknikerns syn på vad en verksamhet är, eller med teknikerns självsäkra attityd av vad verksamheten borde och kunde ha nytta av om bara beslutsfattare och ansvariga i verksamheten förstod de avancerade tekniska lösningarnas alla finesser och fördelar.

Intressant att notera för övrigt var det snabbt växande intresset kring komponent-begreppet. Komponenttänkandet innebär en gradvis förskjutning i synen på tillämpningar från distribuerade men homogena lösningar till mer dynamisk integrering av beständig eller temporär funktionalitet för uppkomna syften. Distribution baserad på treskiktssprincipen och med hjälp av CORBA/DCOM kan realiseras idag, representerar eförnyelse i tänkande och i lösningar men är knappast längre trendintressant. Att sammanställa tillämpningar från fristående komponenter är att gå motsatta vägen. Beteckningen virtuella tillämpningar används ibland. Här saknas ännu kunskap, redskap och erfarenheter. Att metodiken till vissa delar väl svarar mot ett normalt industriellt

produktionstänkande borde kunna öka både attraktionskraft och förmåga. Överlag tycktes många nu betydligt mer optimistiska vad gäller möjligheten att ersätta ”nybyggarentusiasmen” med ingenjörsmässighet. Kanske kommer till sist makten över IT-stödet att hamna hos de verksamhetsansvariga.

Kanske är också tiden välfunnen för en förnyelse av Object World. Låt oss hoppas att nästa års konferens under rubriken ”COMDEX/Enterprise” inte bara till namnet utan även i realiteten innebär ett avstamp mot mer verksamhetsbaserad konstruktion och tillämpningsmiljöer, mot den förnyelse förhandsinformationen utlovar.

Avslutningsvis, det bestående budskapet från konferensen levererat av inledningsanförare Landry när han talade om nya infrastrukturer som de globala näten, ny typ av middleware som CORBA/DCOM och ett ökat fokus på komponentbaserade system: ”We are emerging a new level of computing.” Don’t be left behind.”