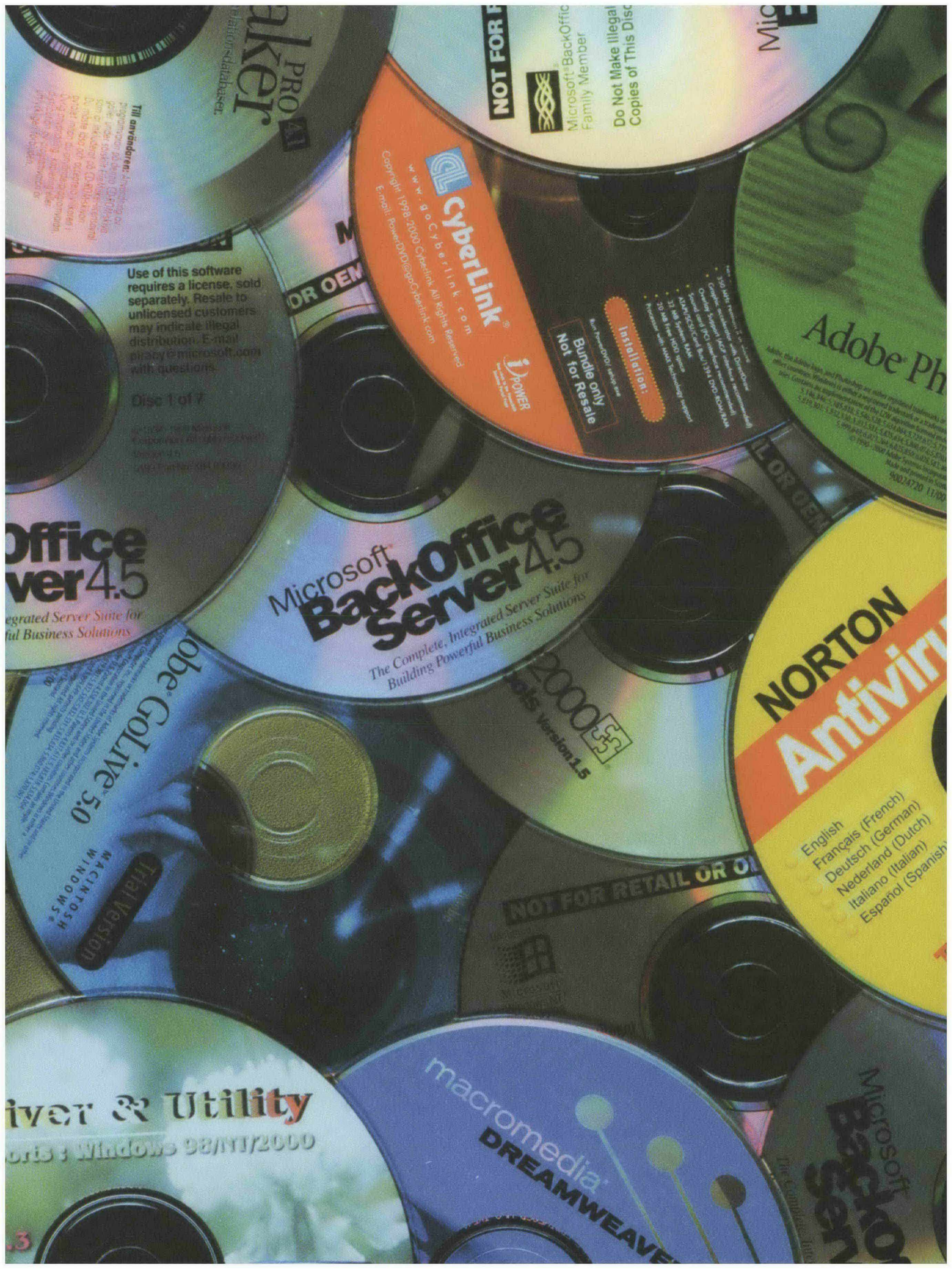

Use of this software requires a license, sold separately. Resale to unlicensed customers may indicate illegal distribution. E-mail piracy@microsoft.com with questions.
Disc 1 of 7
Microsoft BackOffice Server 4.5
The Complete, Integrated Server Suite for Building Powerful Business Solutions
CyberLink
Bundle only Not for Resale
Installation
Do Not Make Illegal Copies of This Disc
NOT FOR RETAIL
NORTON
Antivirus
English
Français (French)
Deutsch (German)
Nederland (Dutch)
Italiano (Italian)
Español (Spanish)
macromedia
DREAMWEAVER

YNGVE SUNDBLAD

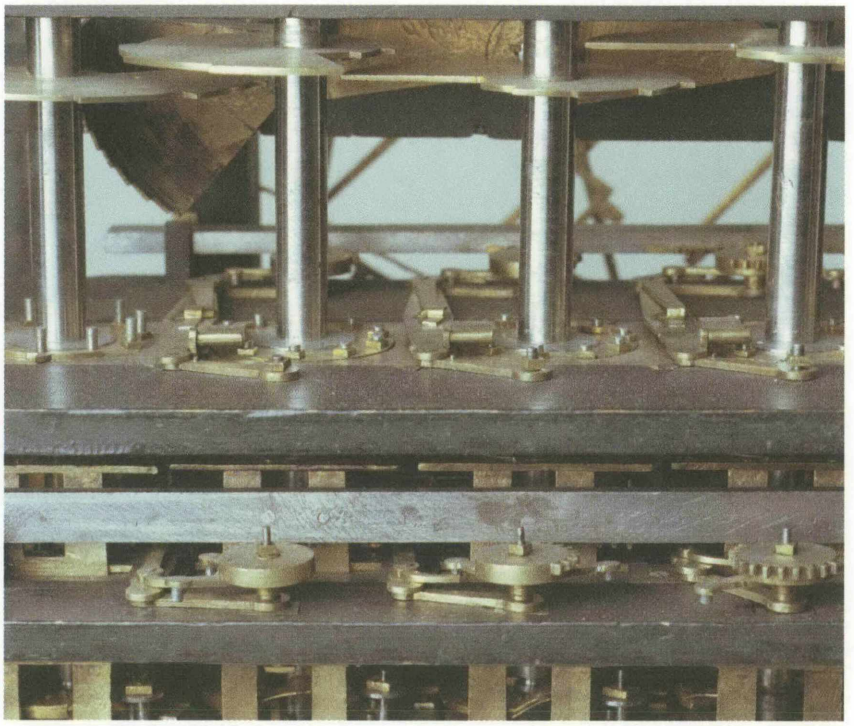
Datorprogram

Räkneverktyg för några blir informationsverktyg för alla

Datorhistorien är kort jämfört med traditionen i de flesta andra mänskliga verksamheter. Det är bara 55 år sedan den första datorn, ENIAC, byggdes vid University of Pennsylvania. För 45 år sedan kommunicerade specialister med datorer med siffror och stora bokstäver i endimensionell text, främst för tekniska och vetenskapliga beräkningar. För 35 år sedan började bredare användning för administration. För 25 år sedan kom den första persondatorn. För 15 år sedan började tvådimensionell grafisk interaktion komma i mer allmän användning. Under de senaste 10 åren har datorer börjat användas för kommunikation och intellektuella upplevelser inte bara med text och grafik utan också video, ljud, animering och tredimensionella världar i det exempellöst snabbt expanderande världsomspännande datornätet.

Samtidigt måste man ha i minnet att inom de flesta områden, inte minst media, där datorer används, finns en lång, ofta månghundraårig, tradition av uttrycksformer och arbetssätt, som måste tas tillvara också vid användning av datoriserade verktyg om man ska kunna uppnå lika goda, eller till och med bättre, resultat och produkter.

En liten del av programutbudet i en butik
FOTO NISSE CRONESTRAND, TEKNISKA MUSEET



Detalj av differensmaskin gjord av svenskarna Edvard och Georg Scheutz, samtida med Charles Babbage. L 1872.
FOTO NISSE CRONESTRAND, TEKNISKA MUSEET

Datorn och människan

Även om datorutvecklingen gått exempellöst snabbt kommer vi se att det även här går väsentlig tid mellan visionär (även "demonstrerad") innovativ datoranvändning och att denna kommer i allmänt bruk. Påtagligt är också att de stora innovationerna gjorts av datorkunniga visionärer som kompletterat sitt teknikkunnande med humanistiska och sociala perspektiv, ofta i samarbete med andra, såväl mediakunniga som användare.

En stor del av dagens datorprogram används direkt i vardagslivet. Avgörande för användbarheten är utöver *funktionaliteten*, dvs vad programmet kan göra, utformningen av *interaktionen* människa–dator. *Användargränssnittet* mot datorn, dvs presentationen på skärm (och eventuellt högtalare) och interaktionsverktyg för att styra datorn, är ju det som användaren upplever som "datorn".

Allra mest explosiv har utvecklingen av datorutrustningens prestanda i förhållande till pris och storlek varit. Detta och innovativa idéer och utrustning för interaktion människa–dator har möjliggjort nya tillämpningar inom snart sagt alla mänskliga verksamheter. Metodik och teknik för att

programmera datorer har, trots de många innovationer som beskrivs nedan, blivit en begränsande faktor. Även dagens mest avancerade programmeringsteknik kräver mänsklig kreativitet och hantverk som tar tid om man skall tillgodose användarnas rättmätiga krav på goda verktyg.

I denna artikel tar jag i första hand upp program och programmering som syftar till direkt mänsklig datoranvändning men vill här påpeka att datorprogram som styr och anpassar tekniska system utan direkt mänsklig inblandning också har en enorm spridning. Dagens tåg, bilar och flygplan liksom dagens telekommunikation, inte minst den mobila, och processindustri är helt beroende av sådana datoriserade komponenter och styrsystem. Den programmeringsteknik som används i detta sammanhang är den som också etablerats för interaktiva tillämpningar, men ofta med särskilt höga krav på tillförlitlighet och säkerhet, vilket nödvändiggör omfattande testmetodik.

Ett datorprogram är, mer eller mindre direkt, en modell av den värld (tillämpning) som ska hanteras. Som vi kommer att se nedan har många alternativa sätt att modellera, som passar bra i olika typer av tillämpningar, utvecklats genom åren.



Ada Lovelace var medarbetare till Charles Babbage på 1830-talet och räknas i vida kretsar som världens första programmerare. Hon har fått ge namn åt programspråket Ada, lanserat kring 1980. Gravyr av WH Mote 1838 efter teckning av AE Chaton RA.
BILD FRÅN ADA PICTURE GALLERY

Datorvisionärer

De flesta av idéerna och begreppen kring dagens datoranvändning med grafiska gränssnitt och datornätanslutna persondatorer presenterades som idéer av visionärer decennier innan de kom i allmänt bruk. En anledning till detta är att idéer behöver "mogna". En annan uppenbar anledning är att den datorutrustning, som behövs, först långt senare kunnat masstillverkas till ett måttligt pris.

Det första exemplet är datorn i sig, en beräkningsmaskin vars operationer styrs av ett lagrat program, som beskrevs som "The Analytic Engine" och delvis realiserades med finmekanik av Charles Babbage 1864. Först när den kunde realiseras elektroniskt och när John von Neumann utvecklat Babbages idé till att lagra program och data i samma binära form, började datorutvecklingen på allvar under senare delen av 1940-talet.

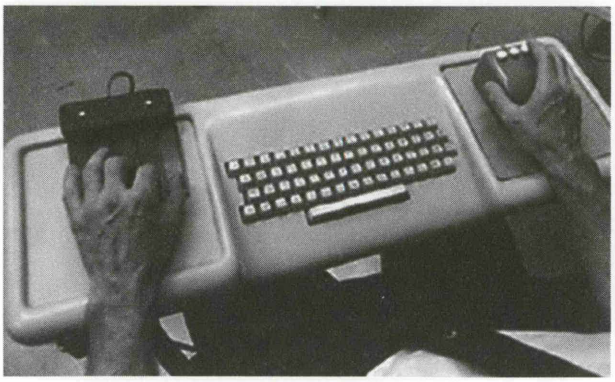
Ett tidigt exempel är idén med hypertext, automatiska hjälpmedel för att få fram ett textstycke som refereras till i annan text. Den beskrevs 1945 av Vannevar

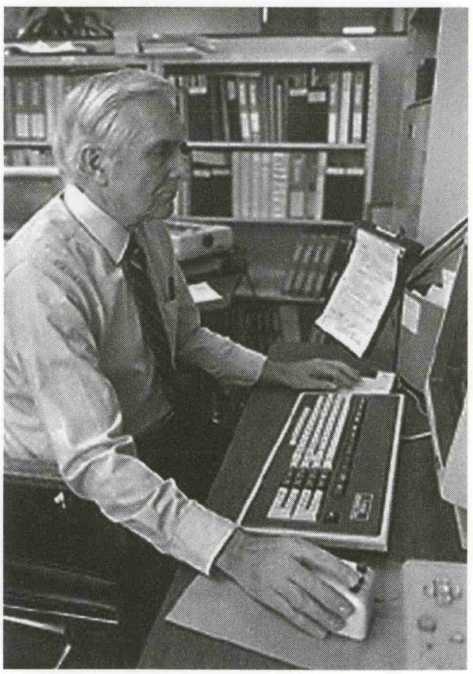
Bush, president Roosevelts rådgivare i forskningsfrågor. Vannevar Bush tänkte sig detta realiserat med microficheläsare och mekanik som automatiskt plockade fram microfichekortet med den aktuella texten. Denna idé, beskriven innan den allra första datorn konstruerats, demonstrerades på datorskärm 1968 (se nedan) men kom i mer allmänt bruk först genom HyperCard på Macintosh 1986 och i exploderande massanvändning i form av World-Wide-Web från 1993.

I en doktorsavhandling från 1963 presenterade Ivan Sutherland Sketchpad, datorstöd för ritande med ljuspenna på en bildskärm där han åstadkommer figurer med gummibandslinjer, gravitation (objekt som fäster vid varandra när de kommer nära), hierarki av figurer och delfigurer, m m. Ritprogram började utvecklas kommersiellt under 1970-talet men fick genomslag först med de grafiska arbetsstationerna i mitten av 1980-talet.

Programspråket Simula utvecklades i Norge av Ole-Johan Dahl och Kristen Nygaard och blev sedan det presenterades 1967 "modersmål" för min generation av akademiska datorprogrammerare i Skandinavien. Simulas innovativa begreppsapparat, där man beskriver det som behandlas i form av *objekt* med *attribut* och *procedurer (metoder),* kom i allmänt (och numera dominerande) bruk som *objektorienterad programmering* först under 1980-talet.

Vid ett framträdande i San Francisco på Fall Joint Computer Conference 1968 demonstrerade Douglas Engelbart, via en radiolänk till en dator i laboratoriet i Menlo Park (sex mil därifrån), Augment, en datormiljö som innehöll det mesta av funktionaliteten i dagens persondatorer och arbetsstationer. Detta var resultat av fem års forskning och utveckling vid Stanford Research Institute. Han visade *ordbehandling, mus, fristående*





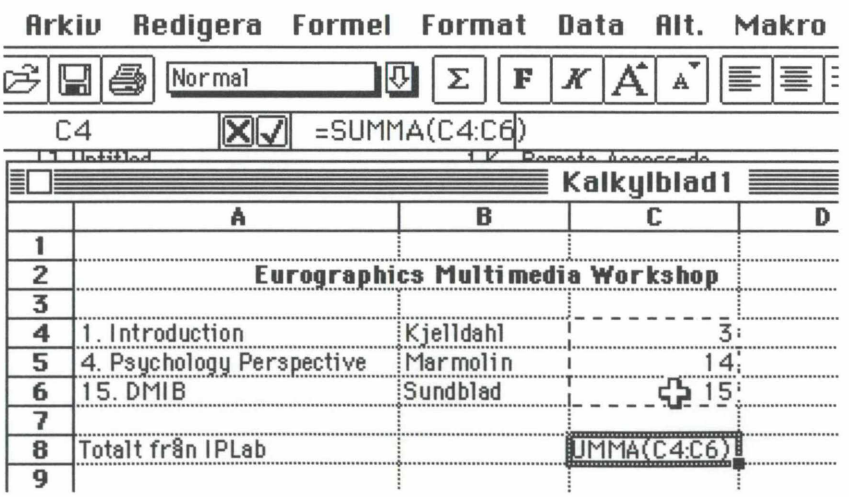
(Nedan t v) Douglas Engelbarts interaktionsverktyg, ackordstangenter, traditionellt tangentbord och mus på en sväng- och tippbar arm för bekväm arbetsställning.
FOTO STANFORD RESEARCH INSTITUTETS ARKIV

Engelbart vid datorn.
FOTO SRIS ARKIV

de tangentbord, fönster, klipp-och-klistra, hypertext, datorvideokonfererens, bekväma datoranvändarmöbler m m. Det tog ca 20 år för flertalet av dessa datoranvändningsformer att komma i allmänt bruk.

I sin doktorsavhandling från 1969 beskriver Alan Kay sin vision om den "dynamiska boken", DynaBook, en bärbar dator med skärm med god upplösning och en penna för interaktion med text-, bild-, video- och ljudmaterial. Han ledde sedan tillsammans med Adele Goldberg under 1970-talet en grupp på Xerox PARC (Palo Alto Research Center) som uppfann och utvecklade många av de begrepp och verktyg som behövs för att möjliggöra DynaBook: *överlappande fönster, ikoner, objektorienterad programmering* och *presentation, pop-up-meny, "cut-and-paste" (klipp-och-klistra), "drag-and-drop" ("dra-och-släpp")*. Under 70-talet utvecklade Xerox PARC andra komponenter, som är självklara i dagens datormiljö: "högupplösande" *bildskärmar med rastergrafik*, idén att representera skärminnehållet som en *bitkarta* eller snarare *pixelkarta* (pixel står för picture element) i datorns arbetsminne, *laserskrivare, sidbeskrivningsspråk* och *Ethernet*.

En av datateknikens bästa uppfinningar gjordes 1978 när två Harvardstudenter inom datateknik och ekonomi, Bricklin och Franklin, insåg att man genom att utnyttja möjligheten att skriva var som helst på skärmen kunde skapa ett dynamiskt *kalkylark* med fält som beror av en instruktion, t ex är summan av andra fält, och som uppdateras direkt vid förändring. Här gick det snabbare. Deras VisiCalc blev en stor försäljningsframgång i början av 1980-talet, men den stora marknaden för denna briljanta typ av



Ett kalkylark
SKÄRMUTSKRIFT AV FÖRFATTAREN

datorprogram, med vid användning, togs så småningom av Microsoft med det idag helt dominerande Excel. Excel är också intressant som programmeringsmiljö för enkla tillämpningar.

Xerox försökte kring 1980 lansera den grafiska arbetsstationen som kontorsverktyg i form av Star. Här finns ett genomarbetat användargränssnitt i form av skrivbordsmetafor, systematisk ikonanvändning och *WYSIWYG* ("What-You-See-Is-What-You-Get"). Genom den specialbyggda hårdvaran blev den så dyr att bara ett fåtal exemplar fick avsättning (på några chefsskrivbord). Allmän spridning fick sådana datormiljöer först från 1984 med den med standardkomponenter byggda Apple Macintosh.

Betydligt snabbare har *World-Wide-Web*-idén spritts från Tim Berners-Lees uppfinning vid CERN 1990 till exploderande massanvändning 1993. Här är det uppenbart att idén med en Web-läsare, Mosaic (numera Netscape) med ett genialt enkelt användargränssnitt, av Marc Andreesen m fl vid University of Illinois, var en avgörande faktor.

Efter dessa nedslag i visionära höjdpunkter gör vi nu en genomgång av hur strukturer, miljöer och arbetsformer för programutveckling utvecklats parallellt med datorernas nya användningsområden under drygt fem decennier.

Programstrukturer och programspråk

De grundläggande strukturer som uppfunnits för att ge programmerarna uttryckssätt har inte blivit inaktuella utan byggts på genom åren. En utveckling av alltfler verktyg har vuxit fram genom att datorutrustningens kapacitet möjliggjort nya tillämpningar.

De som programmerade de första datorerna, främst för tekniska beräkningar, tvingades uttrycka sig i form av *maskininstruktioner* och data i rent binär (nollor och ettor) form. Mycket snart införde man dock mer lättmemorerade symboliska (alfabetiska) skrivsätt för instruktionerna, som sedan översattes (assemblerades) till binär form av datorn själv. I viss utsträckning används denna teknik forfarande när man vill programmera systemdelar, som är så extremt tidskritiska, att datorns snabbhet måste utnyttjas maximalt.

En tidig insikt var att programmen naturligt byggs upp av tre *styrstrukturer*:

- *sekvens* (att göra instruktioner efter varandra)
- *val* (låta ett villkor styra vilken av två instruktioner som blir nästa)
- *slinga* (upprepa en instruktionsföljd ett antal gånger).

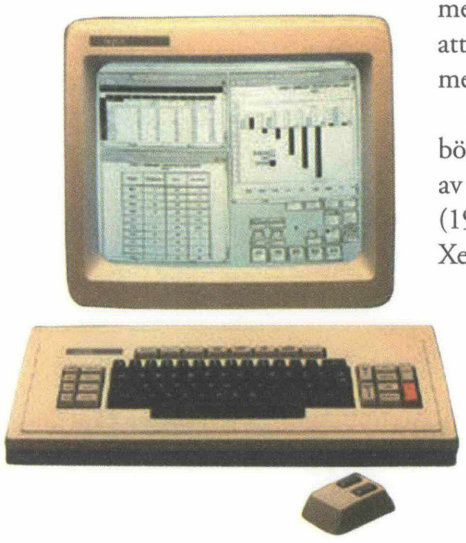
Man kan bevisa att dessa tre strukturer räcker för att bygga upp allt som rimligen kan definieras som databehandling och de går igen direkt eller indirekt i alla datorprogram.

I England uppfanns 1948 idén med *subrutin*, ett programavsnitt som kunde anropas från andra program och åstadkomma och returnera resultat. Den används fortfarande flitigt, ibland under namn som *procedurer* och *metoder*.

Kring 1960 utvecklades den textmässiga beskrivningen av vad datorn förväntas göra till "högre" *procedurorienterade* språk med alltmer kraftfulla grundoperationer. Där beskriver man i textform de operationer som ska utföras. De första standardiserade programspråken var Fortran och Algol för beräkningstillämpningar och Cobol för administrativa tillämpningar. Fortran blev dominerande fram till ca 1980. Basic fick stor spridning, inte minst med minidatorernas intåg under 1970-talet och i gryende användning i utbildning. Pascal, en vidareutveckling av Algol, nådde stor spridning eftersom det med måttligt programmeringsarbete kunde installeras på många olika typer av mini- och persondatorer.

Ökad användbarhet

För den som använder ett interaktivt datorsystem är det naturligt att betrakta det man ser på skärmen som objekt. *Objektorienterad programmering* är en metod att modellera den värld ett program ska hantera och har visat sig vara mycket fruktbar vid utveckling av visuella tillämpningar för arbetsstationer. Objekten förses med såväl *attribut*, dvs datastrukturer, som *metoder*, dvs handlingsmönster (procedurer). Programmets utförande består i att objekt mottar meddelanden om att en viss metod ska utföras, vilket i sin tur kan bestå i att meddelanden skickas till andra objekt.



Xerox Star.

Utvecklingen av objektorienterad programmeringsteknik började i Norge med programspråket Simula (1967). I resten av världen började dessa idéer spridas först med Smalltalk (1980), utvecklat av Adele Goldberg och medarbetare vid Xerox PARC. I Smalltalk är objektbegreppet konsekvent genomfört. Allt är objekt: tal, texter, grafiska objekt på skärmen, hjälpfönster för överblick m m.

Objektorientering har fått en dominerande roll, speciellt vid programmering av interaktiva datortillämpningar. Ur denna tradition kommer Java, ett av dagens dominerande programmeringsverktyg.

Fönster är ytor på bildskärmen. De kan överlappa varandra och fyllas med information som blir tillgänglig samtidigt. Fönsterteknik kan alltså motverka de överblicksproblem skärmens måttliga storlek ofta medför. Fönster används inte bara för presentation utan också för interaktionsobjekt, t ex ikoner (små bilder), dvs symboler som representerar operationer eller data och kan väljas genom markering med pekverktygsstyrd markör, för textmenyer och för paletter uppbyggda av ikoner. Alla som använt en dator idag känner igen tekniken.

Redan 1968 demonstrerade Engelbart, som nämnts ovan, en primitiv fönsterteknik men ett systematiskt sätt att hantera fönster kom först med *de grafiska arbetsstationerna*, som utvecklades i laboratorier som Xerox PARC kring 1976 och kom i allmännare bruk med Macintoshen 1984. Datorskärmens bildpunkter representeras som en *bitkarta* i snabbt arbetsminne i datorn. Detta möjliggör att man kan programmera grafiska gränssnitt med rörelser i "realtid" av musmarkör, fönster, fönsterinnehåll m m.

Datorns operativsystem, det program som alltid finns aktivt, programmeras att styras av "händelser" *(events)* som knapptryckningar, musrörelser, diskettinmatningar m m. Denna teknik användes redan från början på Macintoshdatorer och är grundläggande för funktionen hos dagens interaktiva gränssnitt också för t ex Windows.

Både procedurorienterad och objektorienterad programmering är *imperativ* i meningen att man beskriver instruktion för instruktion vad som ska göras. Som komplettering kan man programmera *deklarativt* i form av *beroenden* ("constraints") mellan data i modellen och lämna till datorsystemet att ta fram den sekvens av operationer som ser till att beroendena upprätthålls. Det är i många tillämpningar väsentligt att kunna uttrycka beroenden mellan objekt på skärmen och/eller i programmet, exempelvis fixerade avstånd mellan spalter på en dokumentsida och mellan väggar och dörrar i en arkitektskiss av ett hus.

Programmeringsmiljöer

Flera av de visionära innovationer som beskrivs ovan gjordes av programmerare för att underlätta sitt eget arbete. De blev de första användarna och snart spreds användningen till andra programmerare och till andra användare för skapande arbetsuppgifter.

Kalkylark med programmeringsmöjlighet i form av att låta fält bero av varandra är ett så kraftfullt hjälpmedel att det ofta kan ersätta utveckling av traditionella program, inte minst i arbetslivet, såväl på kontor som på verkstadsgolvet. Det finns många exempel på "egenutvecklade" Excelark,

där de som kan arbetsprocessen åstadkommit mycket bättre planeringsstöd än centralt installerade standardsystem för tiotals miljoner kronor.

För utveckling av moderna grafiska interaktiva tillämpningar är Smalltalk en stilbildande programmeringsmiljö som blev mer allmänt efterfrågad och spridd i och med att grafiska arbetsstationer blev tillgängliga i början av 1980-talet. I princip all programkod i hela systemet är tillgänglig och möjlig att förändra och testa genom att klippa, klistra, markera och utföra.

Reell användarmedverkan redan från början i en interaktiv datortillämpning underlättas avsevärt av tidiga prototyper som illustrerar hur användarytan kan tänkas se ut och manipuleras åtminstone i några typiska fall. Erfarenhetsmässigt är det oftast mer värdefullt att få fram prototypen tidigt även om den är i primitiv form.

Ett tidigt verktyg för att utveckla grafiska gränssnitt är det objektorienterade HyperCard (1986), som användes mycket på Macintoshdatorer i slutet av 1980-talet. Idag är Director, ett kraftfullare verktyg, tillgängligt också för pc. Många användare, även utan vana vid programmering, har lärt sig hantera HyperCard eller Director relativt snabbt vilket ytterligare förstärker deras möjligheter att prova olika idéer vid deltagande i programdesignprocessen. Idag kan även enkla prototyper åstadkommas med verktyg för att skapa webbtillämpningar, varigenom prototyperna blir tillgängliga för användare spridda geografiskt.

Programmeringsarbetets organisering

Programmeringen var i början, på 1950-talet, ett rent hantverk och även med dagens kraftfulla programmeringsverktyg finns det delar, i varje fall vid utveckling av nya typer av tillämpningar, som kräver hantverksmässig yrkesskicklighet.

Yrkeskategorin programmerare, med allmänna kunskaper om datorer och specialistkunskaper i programspråk, uppstod när datoranvändningen nått en viss volym kring 1960. Kring 1970 började man tala om en programvarukris. Arbetet och kostnaden för att programmera en tillämpning blev alltmer dominerande över kostnaden för själva utrustningen. Tidens försök till åtgärd var industrialisering av programmeringshantverket, bl a baserad på *strukturerad programmering*, som utgick från att funktionen hos det önskade programmet i varje detalj kunde utvecklas "top-down" från en översiktlig version genom successiva förfiningar, i moduler som kunde fördelas mellan programmerarna.

Med persondatorerna från 1975 och de grafiska arbetsstationerna och dessas programmeringsmiljöer kom en insikt att strikt tillämpning av top-

down-metodiken kan hämma kreativiteten i utvecklingen. En modifierad form, *strukturerad tillväxt*, där man omväxlar med att bygga på inifrån för att snabbt få fram prototyper kan ge dels mer innovativa resultat, dels möjlighet att involvera användarna på ett tidigt stadium. Hjälpmedlen måste gynna kreativitet och fantasi och avlasta bokförings- och kontrollarbete. Modern utrustning med möjlighet till avancerad interaktion gynnar ett kreativt arbetssätt, med paradexempel som Smalltalk från Xerox PARC som nämnts ovan.

Den metodik som används av avancerade programmerare vid utveckling av nya tillämpningar med hjälp av grafiska arbetsstationer är i hög grad explorativ och experimentell. Erfarenheterna vid användning av t ex Smalltalk och Java är att man i mycket stor utsträckning använder och ändrar i existerande programdelar. Ett centralt problem är att lära sig hitta det man kan utnyttja i den rika miljön.

Bland oberoende programmerare, inom och utanför universiteten, finns en tradition av gemensam utveckling av fri programvara, som frambringat ett antal goda och mycket spridda program, från operativsystem (grundprogram som styr datorns verksamhet) till textredigerare, programutvecklingsmiljöer, programöversättare m m.

Ett exempel är "Free Software Foundation", sammanhållen av Richard M Stallman, som sedan 1970-talets början genom frivilliga insatser från tusentals programmerare runt världen utvecklat ett stort antal grundläggande verktyg för Unix-miljö. Dessa är ofta bättre än de som säljs av leverantörer och driver också på dessa.

Ett färskare exempel från slutet av 1990-talet, är finnen Linus Torvalds operativsystem Linux, som framgångsrikt konkurrerar med Windows och Unix för persondatorer.

Erfarenheterna från denna samlade utveckling och av ett antal lyckade och mindre lyckade programmeringsprojekt har insiktsfullt samlats till en praktik som i slutet av 1990-talet lanserats under namnet XP (Extrem Programmering) av ett antal utvecklare som också varit pionjärer inom objektorienterad metodik. Fortfarande är behovet stort av metodik som balanserar mellan hantverk, kreativitet och produktivitet. Programvarukrisen består.

Kortfattat består XP av att man "extremt", dvs konsekvent och tätt återkommande, tillämpar följande praktik.

- Gör en grov plan baserad på affärsprioritet och teknik. Uppdatera när man vet mer.
- Sätt ett litet system i produktion snabbt, släpp nya versioner med korta tidscykler

- Vägled utvecklingen med en enkel gemensam metafor (bild av systemets funktion)
- Utforma så enkelt som möjligt. Ta bort extra komplexitet när den upptäcks.
- Programmerare utformar tester som utförs efter varje programförändring. Användare utformar tester som demonstrerar att en viss önskad egenskap fungerar.
- Programmerare omstrukturerar systemet utan att ändra dess funktion genom att ta bort duplikat, förbättra kommunikation, förenkla och lägga till flexibilitet.
- All programkod produceras av två programmerare vid en dator.
- Alla i programmeringslaget kan ändra all kod överallt närsomhelst.
- Systemet integreras många gånger per dag, varje gång en ny del är klar.
- Ingen arbetar som regel mer än 40 timmar i veckan och aldrig övertid två veckor i rad.
- Inkludera verkliga användare i gruppen, tillgängliga jämt för att svara på frågor.

Tillämpningsområden

Tekniskt-numeriska tillämpningar (1950–)

De första datortillämpningarna var rena beräkningar av tekniska och naturvetenskapliga förlopp och konstruktioner. Denna användning av datorer är ännu mer omfattande idag, kompletterad med de stora möjligheterna att grafiskt visualisera resultaten, men andra mycket omfattande tillämplighetsområden har tillkommit.

Administrativa tillämpningar (1960–)

I början av1960-talet insåg man att datorer också kunde utnyttjas för administrativa tillämpningar, som snabbt blev de (volymmässigt) dominerande. I den då förhärskande centraliserade stordatorformen med hålkortsinmatning medförde datatekniken uppkomsten av dels en elit av "överstepräster" med exklusiv rätt att programmera och operera, dels ett proletariat med själsdödande datainmatningsuppgifter, såsom stansning, dels en mellangrupp "användare", dvs beställare och mottagare av resultat i form av listor, etiketter, fakturor m m.

Ur ett akut behov av programmeringsverktyg uppstod Cobol som ett första (!) förslag som hängt med sedan dess, inte minst i system som förändrats men inte ersatts med helt nya under många år. Det medförde en ext-

ra efterfrågan på oss Cobol-programmerare från 1960-talet inför de befarade millenniumskiftesproblemen. Den stora volymen administration med datorstöd idag utförs på persondatorer med en blandning av decentraliserade och centraliserade program.

Ordbehandling (1975–)

De första ordbehandlarna var programmerare som från 1960-talet använde primitiva textredigeringsprogram till att skriva och ändra i programtext. Så småningom började dessa program användas också för andra texter, vilket motiverade utveckling av funktionalitet och användbarhet. Spridningen till andra användare tog fart först i slutet av 1970-talet på persondatorer. Ett tidigt program var WordStar som gjordes tillgängligt på persondatorerna, som först såldes i byggsats (Altair 1975) och sedan färdiga, bl a Apple I 1977. Idag dominerar Word denna typ av datoranvändning som är en av de volymmässigt största.

Desk-Top-Publishing (1985–)

Den grafiska persondatorn fick omedelbart stort genomslag som skriv- och ritverktyg, där WYSIWYG, möjligheten att se typografi och form på skärmen (om än inte i lika hög upplösning som på laserskrivar- eller fotosättarutskrift) uppskattades. Snart, kring 1986, kom de första programmen för Desk-Top-Publishing, som först användes av amatörer för att snabbt och billigt sätta ihop text- och bildmaterial till trycksidor för enklare publikationer, och sedan början av 1990-talet utvecklats till verktyg också för professionella grafiska formgivare och producenter. Viktigt i denna utveckling har varit den allt billigare laserskrivaren med allt högre upplösning och språk för beskrivning av sidinnehåll, inklusive typografi, där det dominerande är Postscript. Inom detta tillämpningsområde finns några av de allra bästa datorverktygen såsom PhotoShop för bildredigering, PageMaker och QuarkXpress för att "sätta" trycksidor.

Dataspel (1980–)

En annan tidig användning av grafiska gränssnitt, t o m före dess genomslag i allmän datoranvändning, är elektroniska spel. Spänningen bygger ju såväl på vad man upplever genom att få något presenterat som (i väl så hög grad) att man kan interagera och styra eller åtminstone påverka förloppet i det man ser. Denna styrning utvecklades med pekdonen (vanligt är styrspak, "joy-stick") till *direktmanipulation* där man direkt kan utföra operationer, med omedelbar återkoppling, genom att flytta runt objekt på skärmen. Direktmanipulation har också kommit in som ett bekvämt sätt att

utföra många operationer i användargränssnitt för de flesta tillämpningar. En form är *drag-and-drop* (dra-och-släpp), där man "objektorienterat" tar tag i ikonen som representerar ett material, t ex ett dokument, och startar utförande av operationer på detta material genom att dra-och-släppa denna ikon på en ikon som representerar det program som ska användas. Även idag kan ofta avancerade interaktionstekniker först introducerade i spel inspirera till kraftfullare användargränssnitt också för annan datoranvändning.

Det finns ju ett otal spel, bland de första var PacMan och Tetris.

Interaktiva medier (1990–)

Med de senaste årens utveckling av datorteknik, kommunikationsteknik och mediateknik blir datorer centrala i skapande av mediala (virtuella) upplevelser och gränssnitt. En utveckling av *interaktiva medier* har inletts under 1990-talet.

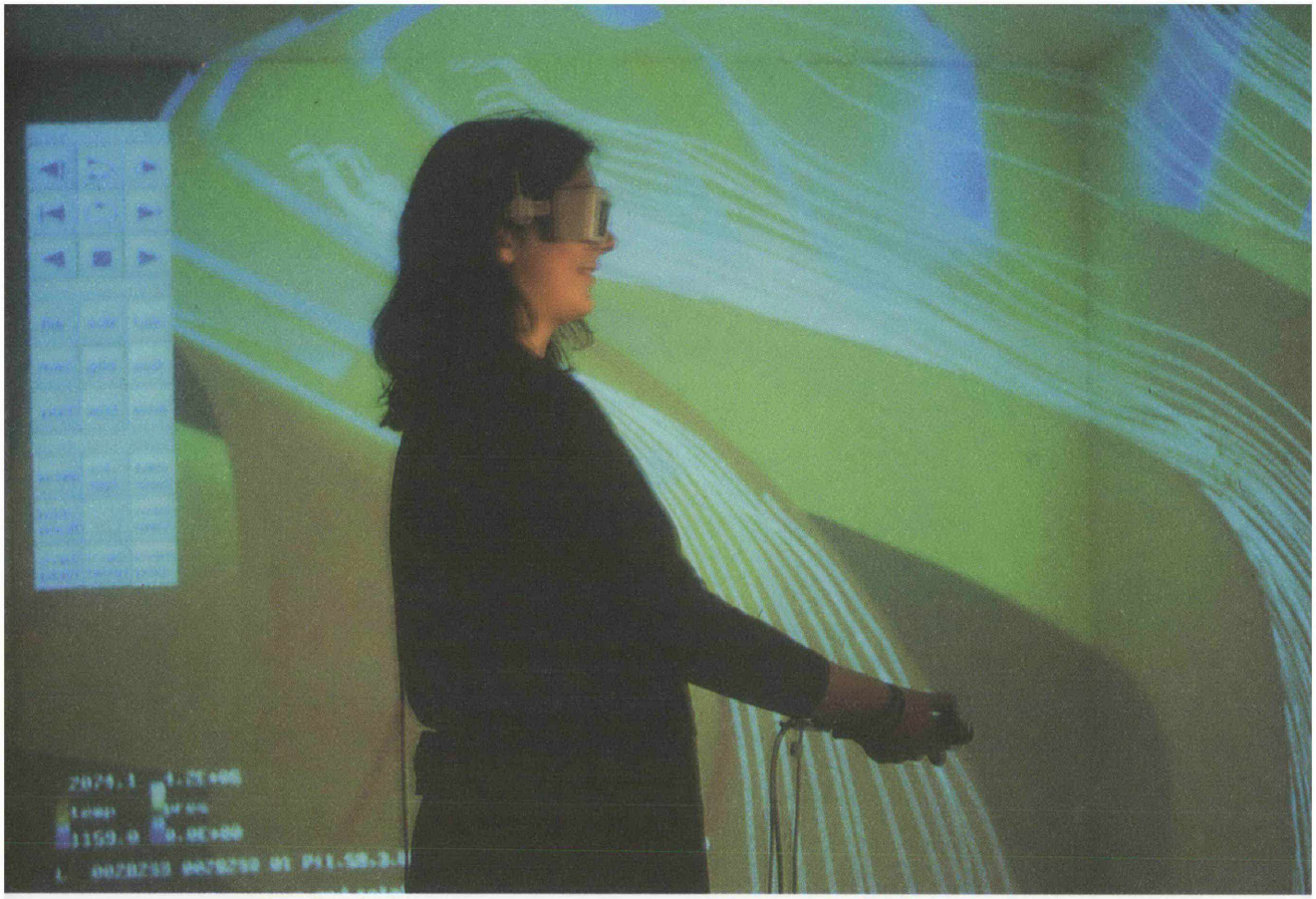
För *ljudmediet* har datorer utnyttjats länge, men bredare användning har kommit först de senaste åren. Elektronmusik skapades med datorer redan på 1950-talet och högtalare har anslutits till datorer för att ge diagnostiska ljud ända sedan datorernas barndom. Nu kommer ljud in också som del av multimediapresentationer och för interaktion, bl a som röstkommandon som kan ersätta eller komplettera grafiska och textgränssnitt.

Video är också intressant som presentationsform i fönster i grafiska användargränssnitt nu när alltfler datorer, även persondatorer, kan utrustas med billiga videokort. Redigering av video görs alltmer digitalt med möjlighet att lätt komplettera med datorgenererad grafik. Interaktiv video har ännu inte funnit sin form.

Mycket använda program för videoredigering är Premiere och Avid och för sammansättning av material till interaktiva mediepresentationer Director.

Kommunikation och information över datornät (1993–)

Extraordinärt i 1990-talets datoranvändning är *datornätens*, speciellt Internets, möjligheter och genomslag för kommunikation och information. Den exploderande massanvändningen av datorpost och *väven* (WWW) gör användargränssnitten extra viktiga. Länge fanns det bara klumpiga datorpostverktyg med textgränssnitt. Kraven var måttliga därför att datorvana hackers var den dominerande användargruppen. Annars hade nog datorpost slagit igenom parallellt med telefaxen redan på 1980-talet. Numera ger t ex Eudora och webläsarna (Netscape och Internet Explorer) bekväm åtkomst till datorpost.



I VR-kuben på KTH
FOTO BENGT VÄNGSTAM, KTH

Ett användargränssnitt som uppenbart hjälpt till att snabbt göra datornäten användbara för breda grupper är Mosaic/Netscape som gjorde Web-uppfinningen lättillgänglig.

Virtuella 3D-världar (1995–)

Datorgenererade (virtuella) tredimensionella världar används i flygsimulatorer och medicinsk utbildning och blir alltmer realistiska i takt med utvecklingen av datorhårdvara såsom specialkretskort för olika grafikoperationer. Kopplat med datakommunikationsteknikens utveckling blir det möjligt att bygga distribuerade världar, där många kan mötas, samverka och dela upplevelser. Här ställs helt nya utmaningar på användargränssnitten, inte minst navigeringsverktyg för tre dimensioner och återkoppling som gör att man inte tappar orienteringen. En mus kopplad till navigeringsfält på skärmen för de tre dimensionerna eller ett "trollspö" (engelska: wand) med vilken man kan röra sig och peka i tre dimensioner, räcker ofta. För spektakulära

tillämpningar finns sensorer på handskar, glasögon, hjälmar med inbyggda bildskärmar för ögonen m m.

I Virtual Reality (VR)-kuben på KTH omges man av på alla sex ytorna utifrån projicerade bilder som skapar en värld där man kan navigera med ett trollspö och uppleva och röra sig i förgångna eller ännu inte byggda (men planerade) stadsmiljöer, 3D-modell av DNA-molekylen, fältet kring en vattenturbin eller ett rymdskepp och konstnärliga installationer.

Programmeringsverktyg för att skapa dessa miljöer är relativt lättanvända och finns också med möjlighet att koppla mellan deltagare som fysiskt är på olika platser.

Närvarande men osynlig datakraft

En parallell utveckling är att datorkraft förekommer i andra former än den traditionella via tangentbord, mus och bildskärm. Några termer för detta är att datorn ses som överallt närvarande *(ubiquitous computing,* Weiser 1991), osynlig *(invisible computer,* Norman 1998) eller försvinnande *(disappearing computer,* som också är namn på ett EU-forskningsprogram 2000). Exempel på centrala tekniker i detta sammanhang är *sensorer* och interaktion via *gester, haptik* (kraftåterkoppling när man vidrör objekt) och *ljud.* Också för detta utvecklas lättanvända programmeringsverktyg.

NÅGRA LÄSTIPS

Följande böcker ger utförligare och kompletterande framställning av temata i denna artikel.

Beck, K (1999) *Extreme Programming Explained.* Addison-Wesley

Borenstein, N (1991) *Programming as if People Mattered.* Princeton University Press

Lammers, S (1990) *De stora programmerarna.* Columna Förlag

Norman, D (1998) *The Invisible Computer.* MIT Press

Raskin, J (2000) *The Humane Interface.* Addison-Wesley

Weiser, Mark, "The computer for the 21st century", i *Scientific American 265 (3), September 1991, s 94–104*