

# SISU *informa*

NUMMER 2 1992

*Nästa generations informationssystem*



## *Användarnas hopp*

*De anpassar datorerna efter människan*



## Stöd för integration i Business Modeller

Business Modeller (BM) utökas just nu med stöd för integration av modeller. Meningen är att man ska kunna använda två stycken BM samtidigt på samma dator och på ett enkelt sätt kunna överföra modeller eller delar av modeller mellan dem. BM kommer också att få ett speciellt verktyg för att navigera i modeller. De nya funktionerna kommer att testas under sommaren och blir tillgängliga för SISUs medlemmar till hösten.

## Svåra ord i stryktipset

Ordet Interoperabilitet (som ni för övrigt får förklarar i artikeln om nästa generations informationssystem, med början på sidan 37) vållade stora problem i förra numrets ordförståelsestryktips. Ingen av de som faxat eller skickat in svar använde det ordet. En förstod det nog, eftersom han hade gjort en speciell kolumn för "Förstår nog men använder inte". Men av det intill nedklottrade "Hu så hemskt" kan man dra slutsatsen att han hellre säger "kommunicerande och samverkande system" än det kortare men svårbegripligare "interoperabla system".

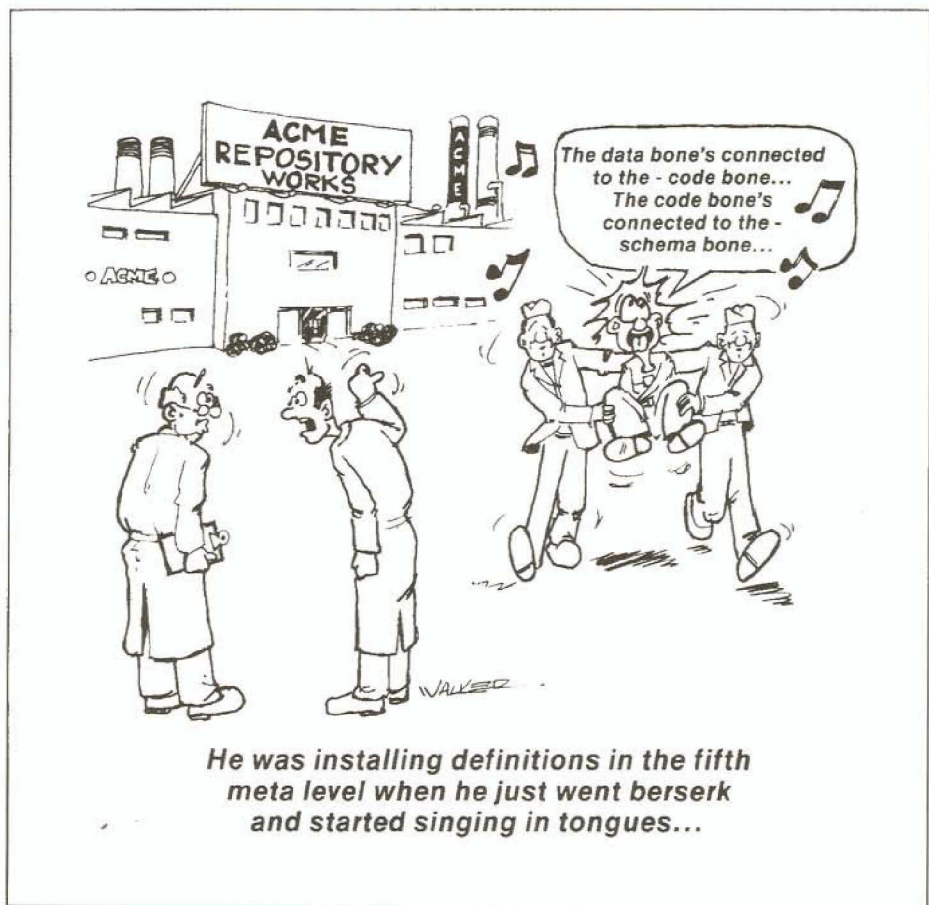
För övrigt kunde inte tipparna enas. Svaren varierade mycket för alla ord. Inget ansågs av alla vara både begripligt och användbart. Men inget uttryck kastades heller av samtliga i papperskorgskolumnen - "varken använder eller förstår uttrycket". Utom Interoperabilitet då. Och ett till, nämligen vårt kontrolluttryck: slamkryparen "Multimodal kodabstraktor". Det förstod ingen. Och tur var väl det, för det gör vi inte själva heller.

## Milord rullar vidare

SISUs AIM-projekt MILORD som startade i januari rullar vidare. Projektet syftar till att ta fram datorstöd för decentraliserade informationssystem i sjukvården med tyngdpunkt på multimedia och kommunikation för gruppstöd. Den första fasen som omfattat förstudier och översikter av projektets olika delområden går mot sitt slut, och en första serie "deliverables" i rapportform har levererats till EG-kommisionen. En workshop med projektets samtliga deltagare hålls på SISU 15 - 17 juni.

Det pågår ett aktivt standardiseringsarbete inom medicinsk informatik, där grupper inom SIS/HSS repre-

senterar Sverige i motsvarande CEN-grupper. Som en del av vårt deltagande i MILORD är det vår ambition att såväl ta del av, som bidra till, standardiseringsprocessen. Ett flertal svenska organisationer deltar nu i AIM-projekt. Samarbete mellan AIM-projekt uppmontras av EGK, och det är SISUs ambition att upprätthålla motsvarande samarbete på "hemmaplan". Ett exempel är GALEN-projektet (Linköpings Universitet), som sysslar med modeller och tekniker för hantering av medicinsk terminologi, något som är viktigt för att kunna bygga informationssystem i allmänhet, och för informationsutbyte i synnerhet.



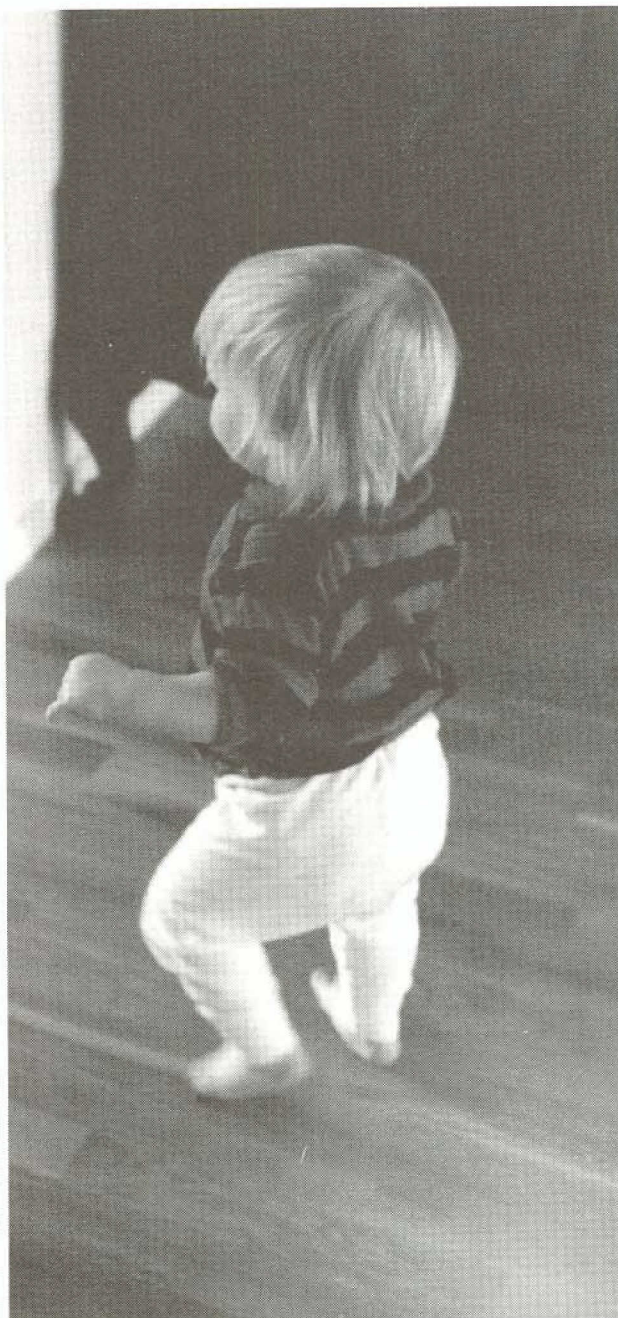
Stämningbild från vår Meta-modelleringsverkstad.

Reprinted with permission from CASE OUTLOOK Vol. 6, No. 1. © CASE Consulting Group, Inc. Lake Oswego, OR. 503-245-6880. Drawing by Lon Walker.



man antagligen först skärskåda sin egen disciplin. Man måste ställa sig frågan om vad det egna området kan bidra med och sedan försöka se de luckor som finns för att till slut hos andra områden finna material att fylla luckorna med. Personligen drevs jag från början av ett grundforskningsproblem som plötsligt fick en ny dimension av att MDI-området kunde förvandla det till ett problem med anknytning till en konkret verklighet. Jag hade som kognitiv psykolog under en tid undrat över människors sätt att tänka logiskt – eller snarare *inte* tänka logiskt. Av en tillfällighet hamnade jag som försöksledare för ett experiment där försökspersonerna skulle söka efter viss information i en databas. Här slogs jag dramatiskt av insikten att det frågespråk som personerna tvingades formulera sina frågor på absolut inte tog hänsyn till människors sätt att resonera. Jag såg en möjlighet att kunna bidra med psykologiska kunskaper till utvecklingen av användargränssnitt för databaser. Men jag såg även gränsen för var mina kunskaper skulle kunna göra nytta och var forskare inom andra discipliner måste ta vid.

Det är engagerande att få delta i och kunna följa den mognadsprocess som framväxten av en ny tvärvetenskap innebär. Processen kan inte påskyndas. Ömsesidig förståelse för olika synsätt är någonting som måste få mogna i sin egen takt. Det är svårt för människor som kommer från olika forskningsdiscipliner att förenas kring gemensamma projekt. Till en början



Människa-datorinteraktion är en tvärvetenskap som tar sina första stapplande steg. Kanske växer området upp till en självständig vetenskap med egen identitet?

Foto: Susanne Sandström/Graffiti.

talas man ju inte ens samma språk. Dessutom råder olika vetenskapsideal inom olika discipliner. Man har olika uppfattningar om hur man bedriver bra forskning och vilka metoder som bör användas. Avsevärd tid måste äg-

nas åt att försöka förstå varandra. Man lyssnar, förklarar och försöker sätta sig in i andra människors sätt att tänka. Detta kan t o m ibland vara smärtsamt, men i de flesta fall samtidigt berikande. Om bara viljan finns till att förstå, tycks också förutsättningarna för att lyckas vara ganska goda. Antagligen finns det en stark vilja. Ett tecken i den riktningen är i alla fall att forskare från olika moderdiscipliner idag genomför gemensamma forskningsprojekt. Det verkar som om kommunikationsbarriären är värd att övervinnas.

Parallellt med att MDI-områdets mångfald av forskare har börjat kommunicera med varandra har även en annan typ av kommunikation börjat utkristallisera sig. Det är en kommunikation som handlar om forskningens tillämpbarhet och som är nödvändig för att MDI över huvudtaget ska kunna bli en tillämpad tvärvetenskap. Utmärkande för MDI är ju utöver dess tvärvetenskapliga karaktär att det är ett område som kräver sk tillämpad forskning – eller kanske mer rättvisande, forskning som är tillämpbar på en praktisk verklighet och på det vardagliga livet. Går det att bedriva MDI-forskning utan att samtidigt ha som mål att bidra till att utformningen av datorsystem blir alltmer anpassade

till de människor som ska använda dem? Jag tror inte det. Idag tycks det finnas en spänning mellan den akademiska MDI-forskningen och de behov som finns i näringslivet. Jag säger *idag*, eftersom det bara är på senare tid



# HYBRIS GÖR DATASERNA BEGRIPLIGA OCH ANVÄNDBARA

NU I WINDOWS-VERSION

**H**ybris hjälper till att skapa ett gemensamt språk i företaget kring information. Tack vare modern teknik för användargränssnitt görs databaserna överskådliga och begripliga för användarna. Med några enkla handgrepp kan vem som helst ta ut den information de behöver i sitt arbete. Hybris finns för Macintosh och Windows 3.

Nu erbjuds alla institutets medlemsföretag möjlighet att använda Hybris, ett program som helt enkelt gör det lättare att dra nytta av alla de investeringar som gjorts i databaser.

En värdefull tillgång i ett företag är dess databaser. Stora investeringar görs för att bygga upp databaserna. Maskin- och programvara är en stor kostnad. Insamling och inmatning av data är en ännu större kostnad. Att underhålla databasen en tredje.



Peter Rosengren, chef för institutets MDI-område.  
Foto: Susanne Sandström.

Men investeringarna i databaserna ger ingen bra avkastning om inte tillräckligt många personer i företaget kan få fram meningsfull information ur databaserna.

Det traditionella sättet att tackla åtkomstproblemet har varit att programmera rutiner som tar fram ett antal standardrapporter. Vartefter användarnas behov ökar utvecklas allt fler rapportrutiner. Efter ett par år sitter utvecklingsavdelningen med en stor mängd standardrapporter som måste vidareutvecklas och underhållas. Trots alla skräddarsydda rapporter har man ändå bara täckt in en bråkdel av vad alla anställda i organisationen egentligen skulle vilja ha ut från databasen. Många luttrade datachefer har upptäckt att *ju mer användarna får veta desto mer vill de veta!*

Ett effektivare sätt att utnyttja de investeringar som gjorts i databaserna vore att utgå från ett fåtal grundläggande standardrapporter och att användarna sedan på egen hand kunde ta ut information ur databaserna vartefter behoven upp-



Den ursprungliga versionen av Hybris utvecklades för Macintosh och används idag på alla teleregioner. Under hösten 1991 och våren 1992 vidareutvecklades Hybris och flyttades dessutom till Windows 3. Detta har skett på uppdrag av Posten.

Windows-versionen av Hybris sattes i drift under april på Televerket. Där används Hybris som ett hjälpmedel för produktionsstyrning. Ett fälttest utförs också för närvarande på Posten mot ett marknads- och säljstödssystem.

En av de grundläggande orsakerna till att Hybris blivit en framgång är att programmet erbjuder ett komplett stöd åt användarna för att komma åt och utnyttja information i databaser. Hybris är inte bara ett frågespråk utan en fullständig miljö för åtkomst av information.

Vid utvecklingen av Hybris utgick vi ifrån att det viktigaste är att ge användarna en överblick över vilken information som finns tillgänglig i databasen och att skapa en förståelse för informationens struktur. Informationen måste göras tydlig och lättförståelig. Kan man göra det kommer också användarna att klara av att formulera sina informationsbehov.

Hybris är modulärt uppbyggt och består av fyra komponenter:

- Selektor
- Uppslagsbok
- Kommunikation
- Skrivbord

*Selektorn* är den grundläggande komponenten i Hybris. Den utnyttjar användarna för att välja ut den information de är intresserade av. För att göra informationen i databasen tydlig och begriplig för användarna finns en verksamhetsorienterad *informationskarta*. Den visar databasens innehåll uttryckt i användarnas egna begrepp och termer, se figur 1.

I kartan kan användarna se vilka viktiga begrepp i verksamheten det finns information om i databasen. Kartan visar också hur olika begrepp hänger ihop.

De som arbetat med modellering inser att informationskartan är samma sak som en begreppsmodell. I Hybris har vi dock valt att kalla det för informationskarta för att betona dess roll som navigeringsinstrument.

Självklart är informationskartan i allmänhet mycket större än vad som kan presenteras överskådligt på en

skärmbild. Det har lösts i Hybris genom att man logiskt delar upp informationskartan i flera delkartor. I varje delkarta samlas en grupp begrepp som ofta utnyttjas tillsammans i vissa frågeställningar. Dessa uppdelningar görs i samråd med användarna.

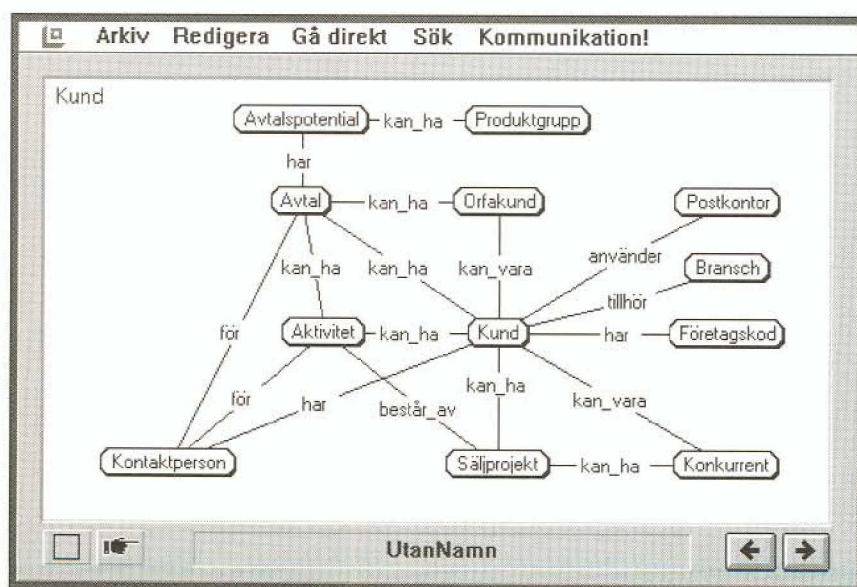
## Peka och klicka

För att välja ut information pekar och klickar användaren direkt i informationskartan på de begrepp som han vill ha information om. Genom ett dubbelklick på ett begrepp öppnas detta och då kan olika sorters villkor formuleras. Öppnar man t ex Kund kan man tala om att man bara är intresserad av kunder i Sollentuna.

I princip all interaktion med systemet bygger på grafik och direktstyrning. Enda gången användarna behöver utnyttja tangentbordet är när de fyller i specifika sökvärden som t ex Sollentuna i exemplet ovan. Dock finns möjlighet för databasproffs att direkt skriva in databasfrågorna i SQL (Structured Query Language).

En grafisk begreppsmodell räcker inte till för att skapa en förståelse för informationen. En mycket viktig komponent i Hybris är uppslagsboken. I den kan användarna direkt slå upp hur olika begrepp, relationer och termer är definierade. Uppslagsboken utnyttjar så kallad hypertext-teknik, se artikel om handboksteknologi i detta nummer av Informa. Det gör det möjligt för en användare att fritt söka sig runt i uppslagsboken för att förstå hur saker och ting hänger ihop i databasen, se figur 2.

Uppslagsboken fylls på med information från datakataloger, existerande begreppsdefinitioner på papper, kodförteckningar et c. På så vis fungerar den också som en lättillgänglig dokumentation av ett system.



Figur 1: Informationskarta med användarens egna termer som orienteringsstöd.



behöver i sitt arbete. Skrivbordet kan i princip skraddarsys på tre olika nivåer:

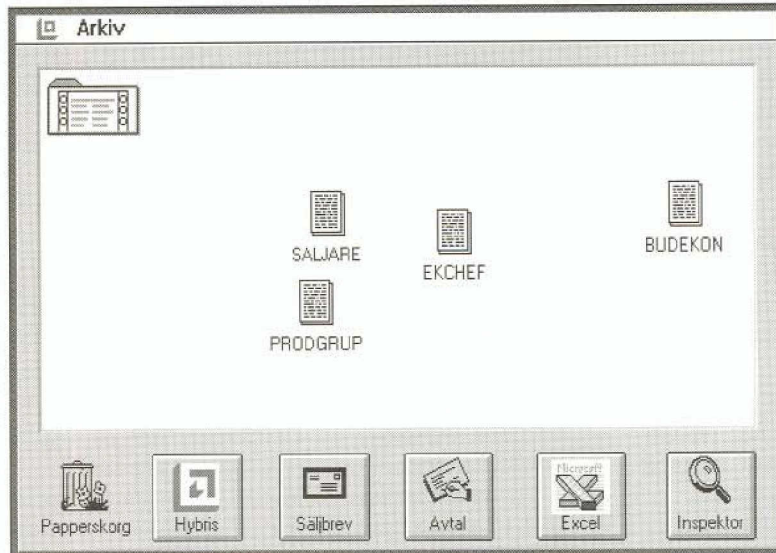
- Företagsnivå
- Tillämpningsnivå
- Uppgiftsnivå

Företagsnivån är en ganska grov anpassning. Då kan man från skrivbordet nå företags standardprodukter, t ex den ordbehandlare och det kalkylprogram som är företagsstandard. En finare grad av anpassning är tillämpningsnivån, vilket innebär

att man bestämmer att till en viss tillämpning hör en viss uppsättning funktioner. Ett exempel är att man slår fast att de som arbetar med kunddatabasen ska ha tillgång till fyra standardrapporter, fem säljbrevsmallar och

två kalkylmakron i Excel medan de som arbetar mot produktdatabasen ska ha sex standardrapporter och ett statistikprogram.

Den mest skraddarsydda nivån är uppgiftsnivån, då man utformar olika skrivbord för olika arbetsuppgifter.



Figur 3: Utgångspunkten för Hybris-användarna är skrivbordet.

Detta kräver metoder för att analysera och beskriva användarnas arbetsuppgifter samt tekniker för att realisera en lämplig uppsättning funktioner för arbetsuppgifterna. Att utveckla sådana metoder och tekniker är föremål för MDI-gruppens fortsatta forskning.

Användningsområdena för ett verktyg som på ett generellt sätt gör det möjligt att söka och utnyttja information från

företagsdatabaser är i princip obegränsade. Dels kan det användas som ett fristående program, dels kan det byggas in i andra program. Ett exempel på det senare är att använda Hybris som ett komplement till existerande EIS-system (Executive Information Sys-

## HYBRIS UTVECKLAS INOM EG

MDI-gruppen arbetar alltid i flera mindre och väl avgränsade steg. När ett forskningsproblem har lösts försöker vi se till att lösningen också kommer till praktisk användning bland våra medlemsföretag innan vi tacklar nästa problem. Detta hellre än att försöka lösa alla problemen på en gång och riskera att aldrig bli färdig.

Hybris är ett exempel på vårt sätt att arbeta. Nu när problemet med fria sökningar i en databas har lösts går vi vidare och tacklar problemen med att kunna söka och hämta data från flera databaser samtidigt. En första prototyp har tagits fram och går under arbetsnamnet *Debris* (Decentraliserad Hybris). Prototypen bygger på idéer från Hybris samt EG-projektet *Kiwis*. Erfarenheterna av denna första enkla prototyp håller nu på att sammanställas och utvärderas.

Ett mycket angeläget problem att komma till rätta med i de flesta organisationer är hur man bygger upp multimedia-databaser, d v s databaser där vanliga data blandas med textdokument, bilder, ritningar, fax, video, kalkylark med mera. En viktig del i framtida multimedidatabaser är hur sökning och åtkomst ska ske på ett för användarna naturligt sätt. Och detta är nästa steg i Hybris-utvecklingen.

Inom EG-projektet *Intuitive* samarbetar SISU med företag från England, Frankrike och Spanien med målet att utveckla nästa generation användargränssnitt för multimedidatabaser. Där ingår Hybristechniken som en viktig komponent. Projektet omfattar 60 manår och har en budget på 70 miljoner, se föregående nummer av *Informa* för mer information.



# THE XANTHUS STORY

VAR DET EN NY SVENSK PROGRAMINDUSTRI  
SOM FÖDDES PÅ KROGEN?

**T**vå unga grabbar som började med tomma forskarhänder och överfulla hjärnor. Två grabbar som arbetade dygnet runt för att de inte kunde tänka sig något roligare. De kom från SISU och SICS, startade eget och håller nu som bäst på att öka Sveriges exportinkomster genom att sälja det egenutvecklade programmet Craftman till andra sidan Atlanten.

Men egentligen började allt på krogen.

Det är lite Silicon Valley över Electrum i alla fall. Ni vet det där om att samla en massa kreativa människor på så liten yta att de inte kan låta bli att stöta på varandra – och att det vid dessa ständiga sammanstötningar frigörs en massa tankeenergi. Erfarenheter utbyts och stora tankar tänks. Allt enligt den mänskliga fysikens lagar. Ibland fungerar det så.

Året var 1990. Jesper Lundh och Björn Backlund arbetade båda med avancerad programutveckling för gra-

fiska användarmiljöer i Electrum, Kista. Jesper på SISU och Björn på granninstitutet SICS (Swedish Institute of Computer Science). Jesper och Peter Rosengren, hans dåvarande chef på SISUs MDI-område (Människadatorinteraktion), gick på föreläsning om distribuerade system hos Björn på SICS. De tre gick på samma konferenser och kurser, stötte ihop på lunchen och så där.

Det ordnades Electrum-mästerskap i brännboll. Institutet för Mikroelektronik, IM, hade fyra egna lag. SISU och SICS tog det hela mindre allvarligt, men fick i alla fall ihop ett gemensamt lag – som mot alla odds vann hela turneringen. Jesper och Björn och de andra i laget gick ut och firade. Och pratade. De fortsatte att umgås och prata över sommaren. Många idéer kläcktes på krogen.

När SISU sedan hade seminarium var Björn där för att demonstrera Next-datorn. De pratade vidare. Båda kände de väl till möjligheterna och begränsningarna i marknadens mest avancerade utvecklingsverktyg. I den nya Next-datorn fanns helt nya möjligheter.

## **Ett gott rykte skadar aldrig**

De sökte och fick bidrag till en förstudie från STU (nuvarande NUTEK, Närings- och teknikutvecklingsverket) sommaren 1990. De fick då råd att

besöka Next och att låta tekniska attachéerna i San Francisco göra en marknadsundersökning.

– Antagligen hade vi ett stort förtroendekapital med oss till STU, tack vare SICS och SISUs goda rykte, funderar Jesper.

De fick båda arbeta med projektet större delen av tiden på SISU respektive SICS. (Projektet hette då GAD – Graphical Application Developer). Parallellt arbetade Jesper i Hybris-projektet. De gick programmeringskurs för Next-datorn.

– Sedan arbetade vi i princip dygnet runt, även helgerna. Ibland tog vi i alla fall en öl-paus på krogen. Vi var ju båda ungar, minns Jesper.

Så höll de på fram till april 1991 då de åkte till Next i USA för att visa upp vad de åstadkommit. Next-folket blev imponerade och uppmuntrade dem att fortsätta, utan att egentligen lova någonting.

## **Innovatörsbidrag och gräddfil**

– Men vi tog steget fullt ut och lade vår framtid i Steve Jobs händer, skrattar Jesper.

(Steve Jobs är en av grundarna till Apple. Riktigt bra för Apple började det dock inte gå förrän Jobs hoppade av och startade Next istället...)

Ja, de satsade verkligen. I maj 1991 slutade de sina trygga och utvecklande forskartjänster för att starta eget före-



Marknaden för Craftman har utökats kraftigt i och med lanseringen av Nexts nya operativsystem för persondatorer; Nextstep 3.0. Från sådär hundra tusen till närmare fem miljoner potentiella användare (Nextstep kan inte bara köras på Motorola-processorer som finns i Nexten, utan även på Intels 486-processorer som sitter i många persondatorer). SISU och SICS får royalty på försäljningen av Craftman de första tre åren.

– Vi har redan börjat utveckla nästa produkt, säger Jesper hemlighetsfullt, och vi har kunnat återanvända en stor del av arbetet vi lagt ner på Craftman. Så det går nog snabbare den här gången.

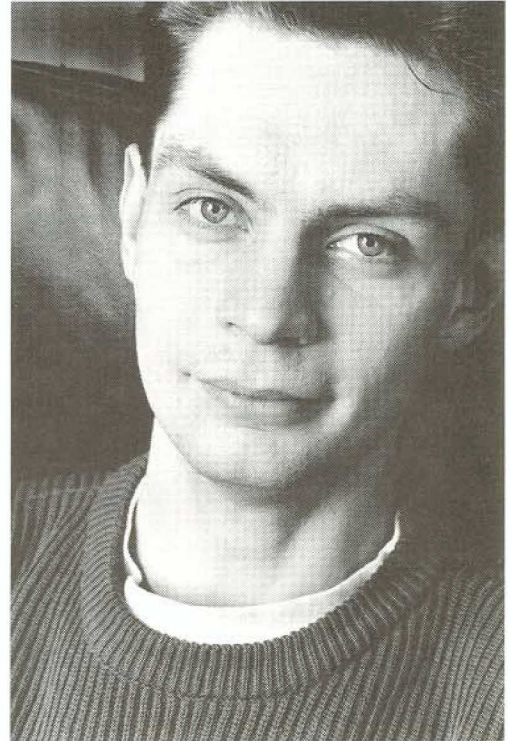
De säger att de gärna vill bli rika som troll och ge upphov till en blomstrande exportindustri för svensktillverkade program. Men andra får gärna bedriva själva företagandet.

– Det är utveckla vi vill, betonar Björn: Även om vi inte fick betalt skulle vi ändå sitta och utveckla program på nätterna. Det är ju vår hobby!

Av Pål Török

*Craftman är ett verktyg med vars hjälp man lätt ska kunna bygga program med grafiska användargränssnitt (det och andra sådana verktyg beskrivs utförligt i artikeln som börjar på sidan 16). Craftman har ett objektorienterat programmeringspråk och utnyttjar Next-datorns stora möjligheter till grafik, video och ljud.*

*Idén som Craftman bygger på är alltså att det finns ett stort behov av företagsanpassade tillämpningar; program som byggs eller ändrats utifrån verksamhetens behov. Med verktygets hjälp går det betydligt fortare att skapa sådana program. Dessutom är det så lättanvänt att även icke-specialisten ska kunna bygga det användargränssnitt som passar just honom.*



Jesper Lundh, tidigare SISU, nu Xanthus.  
Foto: Susanne Sandström.



Björn Backlund, tidigare SICS, nu Xanthus.  
Foto: Susanne Sandström.



skriva att t ex en knapp ska ha en viss storlek, färg och placering kan utvecklaren placera ut knappen, anpassa dess attribut och direkt se hur resultatet ter sig. Utvecklaren behöver inte vänta på resultatet på grund av den vanliga fördröjningen i konventionellt programmeringsarbete, "ändra-kompilera-exekvera". De visuella gränssnittsverktygen ger möjlighet att omedelbart se konsekvenserna av ett designbeslut utan att vänta på tidsödande kompileringar. Det ökar effektiviteten i utvecklingsarbetet.

Ett visuellt gränssnittsverktyg minskar utvecklarens behov av att komma ihåg intrikata detaljer om fönstersystemets och interaktionsobjektens egenskaper. Sådana detaljer kan hanteras av verktyget, som med ett bra gränssnitt kan presentera mycket information på ett överskådligt sätt. Utvecklaren kan exempelvis snabbt få en överblick av tillgängliga interaktionsobjekt genom att de finns i en palett. Gränssnittets struktur kan kanske presenteras som ett träd av interaktionsobjekt, där man ser hur objekten ingår i andra objekt.

### Fyra viktiga komponenter

Följande fyra komponenter är viktiga delar i ett bra visuellt gränssnittsverktyg:

- Ett kodbibliotek, eller en verktygslåda, som antingen kan vara ett externt standardbibliotek på den valda plattformen eller inbyggt i verktyget.
- En layout-editor som enkelt kan visualisera gränssnittet i ett tidigt skede i utvecklingsarbetet.
- Dialogkontroll för att specificera dialogsekvenser och olika interaktionstekniker.
- En analysdel som kan ge råd och påpeka brister redan i konstruktionsfasen. Exempelvis kan analysdelen se till att påpeka avvikelser från den valda gränssnittsstandarderna.



Ulf Wingstedt Foto: Susanne Sandström.

Det finns relativt många verktyg på marknaden som innehåller de första två komponenterna, kodbibliotek och layout-editor, medan dialogkontroll och framförallt analysdel är mer ovanliga.

Vanligen utnyttjar gränssnittsverktygen något kodbibliotek vilket innebär att fördelarna med dessa, till exempel konsekventa gränssnitt (se fakturata), också gäller för gränssnittsverktygen. Men ett bra visuellt gränssnittsverktyg ger ytterligare fördelar:

- Prototyper kan skapas snabbt, till och med före det att tillämpningskoden är skriven, vilket innebär att det är lätt att undersöka alternativa gränssnittslösningar.
- Gränssnitten blir lättare att förändra eftersom gränssnittskoden är bättre strukturerad och, till viss del, separerad från tillämpningen. Det blir lättare att uppfylla användarnas krav.
- Verktyget kan ge tillgång till interaktionsobjekt med större funktionalitet och bättre kvalitet än vad man annars själv kan åstadkomma.

- Personer som inte är datorspecialister kan lättare delta i utvecklingsarbetet, t ex designers och psykologer.

### Hur ska verktyget användas?

Olika typer av programutveckling ställer olika krav på verktygen. Är det en statisk prototyp som ska utvecklas för att visa det framtida programmets utseende? Ska det vara en prototyp med funktionalitet, eller är det det fullständiga programmet som ska utvecklas?

Det finns en mängd verktyg att välja bland om man *enbart vill skapa gränssnittets statiska layout*, dess utseende. I princip alla verktyg som innehåller en layout-editor är lämpliga att använda för detta ändamål.

Från beskrivningen i verktyget genereras sedan ett kodskelett i något generellt programspråk, t ex C. Kodskelettet innehåller den kod som återskapar gränssnittet på skärmen vid körning. I kodskelettet lägger man sedan in anrop till egen kod.

Flera verktyg sparar layout-beskrivningen på ett mellanformat från vilket koden sedan genereras. I exempelvis



gränssnittsverktyg Devguide har t ex i senaste versionen en motsvarande funktion trots att Devguide inte är objektorienterat i egentlig mening.

## Craftman tyglar Next-datorn

Interface Builder är ett verktyg för programmerare medan svenska verktyget Craftman, har ambitionen att ge icke-specialisten tillgång till Next-maskinens möjligheter. Craftman började utvecklas inom ett samarbetsprojekt mellan SISU och SICS och vidareutvecklas nu i avknopningsföretaget Xanthus.

I Craftman har Next-datorn tyglats i en utvecklingsmiljö som influerats av både Hypercard och Interface Builder och som dessutom har ett objektorienterat programmeringsspråk.

En del gränssnittsverktyg baseras på kodbibliotek som underlättar flyttbarhet mellan olika miljöer. De arbetar enligt två olika strategier:

- Utnyttjar de verktygslådor som är standard i respektive värdmiljö och översätter anropen av den egna verktygslådans rutiner till anrop av den lokala verktygslådan. Exempel är XVT från XVT.

- Realiserar interaktionsobjekt för alla värdmiljöer i den egna verktygslådan och utnyttjar inte värdmiljön. Ett exempel på detta är Open Interface från Neuron Data.

Båda sätten fungerar men har lite olika konsekvenser. XVT-program blir mer lika andra program i värdmiljön eftersom de använder samma verktygslåda. Open Interface-program har fördelen att man under körning kan byta gränssnittsstandard, t ex från Motif till Open Look. Det betyder att olika användare kan ha gränssnitt enligt olika standarder.

Det bör nämnas att verktyg som Open Interface och XVT inte underlättar själva gränssnittsutvecklingen första gången, snarare tvärtom. Det är

ofta svårare att utveckla ett gränssnitt som ska vara flyttbart. Verktygen gör det däremot enklare att sedan flytta programmet till en annan miljö.

## UIMS - User Interface Management Systems

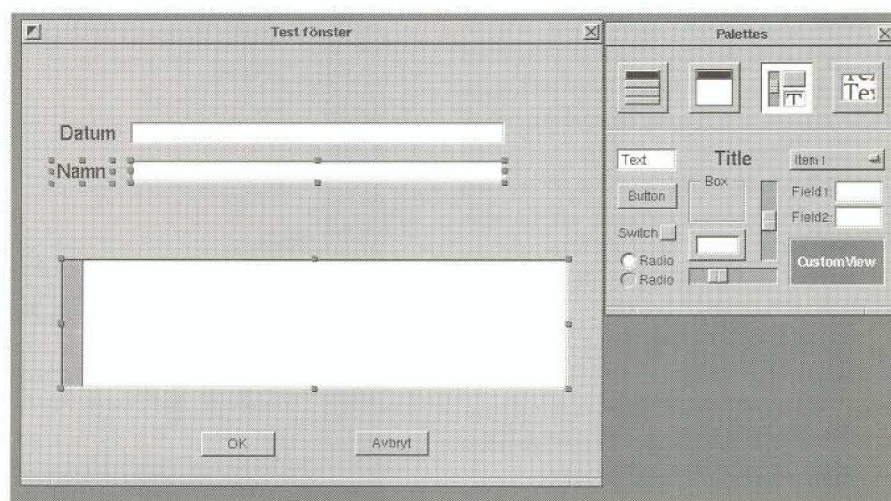
Många av gränssnittsverktygen bygger på att det är möjligt att i viss mån dela gränssnittskoden från den övriga programkoden, eller tillämpningskoden. I enklare former av grafiska gränssnitt fungerar det bra. Från en layout med knappar och menyer genererar verktyget programskelett där det finns ingångar för programmeraren att själv fylla i vad som ska hända när t ex en knapp trycks ned eller ett menyalternativ väljs.

Separationen av gränssnittskod och tillämpning är kungstanken bakom gränssnittshanteringsystemen, UIMS (User Interface Management System). Ett UIMS kan ses som en analogi till databashanteringsystemen, DBMS, d v s man överlåter åt ett UIMS att hantera all kommunikation mellan användaren och programmet på samma sätt som ett DBMS sköter datahanteringen.

Programmen skrivs då mot det virtuella gränssnittet i ett UIMS och blir helt oberoende av den eller de gränssnittsmiljöer där de ska användas eftersom alla skillnader i värdmiljö hanteras av UIMS. Exempel på UIMS är bl a det svenska systemet Teleuse från Telesoft.

Ett UIMS innehåller ofta utvecklingshjälpmedel som till exempel layout-editorer men skiljer sig alltså åt från andra verktyg genom att det är aktivt också under körning av programmet.

UIMS ansågs länge vara en lovande teknik men har under senare år omvärderats något. Moderna gränssnitt med höga krav på interaktion tillåter näppeligen en total uppdelning mellan gränssnittskod och tillämpningskod, speciellt gäller detta program där tillämpningsdata visualiseras, t ex ritprogram. Kontrollen av knappar och menyer kan enkelt förläggas till ett UIMS medan däremot markering av ritade objekt, skalning och flyttning förmodligen måste hanteras av tillämpningen. I många typer av tillämpningar blir uppdelningen oklar och överlappande.



Interface Builder



”villkors-uppfyllare”, försöker det upprätthålla villkoret genom att automatiskt flytta den ena cirkeln om den andra flyttas av användaren.

Eftersom begränsningarna i att beskriva ett gränssnitt i text redan in-setts, har man försökt finna sätt att beskriva villkor med hjälp av visuella tekniker. Ett intressant tillvägagångs-sätt är då att demonstrera för verktyget hur man vill att det ska bete sig. Verktyget genererar sedan automatiskt den kod som behövs för att gränssnittet vid körning ska fungera på avsett sätt. På Carnegie Mellon University i USA har man under ledning av Brad A Myers kommit långt inom villkors-styrda gränssnitt som, till vissa delar, programmeras genom demonstration. Principerna har realiserats i forsknings-prototypen Garnet.

I framtidens verktyg behövs också stöd för alla de nya interaktionsformer som börjar bli verklighet. Penn- och röststyrda datorer finns redan att köpa men det finns inga verktyg som underlättar utveckling av dylika gränssnitt.

Synsättet att man först utvecklar den viktiga tillämpningskoden och sedan lägger på något gränssnitt håller snabbt på att föråldras. Med de nya effektiva verktygen för programutveckling inklusive gränssnittskonstruktion kommer tvärtom själva gränssnittet att bli den viktigaste komponenten i programmet.

Den framtida utveckling av interaktiva program kommer att drivas av utvecklingen av gränssnittet, när gränssnittet är klart är programmet klart! Det enda som fattas är en del generell kod för hantering av exempelvis databaser och kommunikation som enkelt kan länkas in i programmet.

Av Ulf Wingstedt

## Kravlista på verktyg

Med tanke på det stora utbudet av gränssnittsverktyg - hur ska man välja? Förutom egenskaper som den egna lokala datormiljön kräver, är här en lista med några punkter att fundera över:

- Ska prototyper eller färdiga system utvecklas?
- Hur lätt är det att gå från prototyp till färdigt system?

Vissa verktyg passar bättre för prototyper medan andra är bättre för att utveckla det system som ska användas. Om man tänker sig att utveckla prototypen i en miljö och det riktiga programmet i en annan kan det vara lämpligt att undersöka möjligheten till att exportera specifikationen från prototypverktyget till det andra verktyget.

- Inlärningströskel
- Effektivitet i utveckling

För att lyckas med prototyp-utveckling måste verktyget vara effektivt, man ska kunna åstadkomma mycket på kort tid.

- Hur stor del av programmet kan göras i verktyget? Hela, eller bara layout?

För att göra en realistisk prototyp krävs också viss funktionalitet. Det kan vara enklare att ha möjlighet att koda den direkt i verktyget än att tvingas gå utanför.

- Kan programmet underhållas från verktyget? Kan gränssnittet förändras när tillämpningskoden är infogad?

Om kodskeletten fyllts med egen tillämpningsspecifik kod, bör det ändå vara möjligt att ändra programmets gränssnitt med hjälp av verktyget. Det finns exempel där man tvingas skriva in ändringarna i gränssnitsskoden direkt i en vanlig editor.

- Hurstödsstandarder för gränssnitt? Vilka?

Verktyget bör kunna användas för den eller de gränssnittsstandarder som används på företaget.

- Versionshantering?

Under en längre utvecklingsfas är det bra att kunna gå tillbaka till äldre, eller utveckla pararella, versioner vid behov.

- Gränssnittets flyttbarhet?
- Kan olika kodbibliotek användas?

Det kan vara en fördel om verktyget kan användas mot kodbibliotek eller verktygslådor för olika standarder, eftersom det blir lättare att flytta programmet mellan olika miljöer.

- Gränssnittsspecifikationens flyttbarhet?

Alla verktyg har olika egenskaper och en kombination av två verktyg kanske är precis vad som krävs i en viss situation. Att gränssnittsspecifikationen kan flyttas från ett verktyg till ett annat är då ett krav.

- Påverkan av prestanda för färdiga programmet.

Flera av de verktyg som tillåter att hela programmet utvecklas ger ibland sämre prestanda eftersom de inbyggda språken tolkas (interpreteras) vid körning av programmet i stället för att kompileras. Om ineffektiv gränssnittskod genereras påverkar det också prestandan.

- Krävs run-time licens?

Vissa verktygstillverkare kräver att en run-time-licens köps för varje ny kopia av det program som utvecklats med verktyget.



# ELEKTRONISK HANDBOK GER UTRÅDE ÅT SERVICETEKNIKER

punkt för metodutveckling har varit att betrakta den elektroniska handboken som ett informationssystem och att så långt som möjligt använda etablerade tekniker för systemutveckling.

Ett annat mål har varit att utforma ett gränssnitt som stödjer läsaren av handboken i hans arbete. Som ett delmål har här ingått att försöka förstå användarens mentala representation av och orientering i handboken. Vi har sålunda försökt att förena ett systemutvecklingssyfte med mer forskningsinriktade mål. Dessutom tyckte vi att det var viktigt att i detta systemutvecklingsprojekt integrera teknisk kompetens med psykologiska kunskaper.

## Elektronisk handbok på fältet

Eftersom vi ville försäkra oss om att vårt projekt skulle vara förankrat i verkligheten, sökte vi som sammanhang en arbetsplats där man önskade överföra en stor mängd pappersdokumentation till ett elektroniskt format. En sådan arbetsplats fann vi hos Teli i Vänersborg och deras miljö fick därför utgöra ett typpall för vårt projekt. Här arbetar bl a ett antal servicetekniker med att stödja olika telefonväxlar. Deras arbetsuppgifter består till stor del av att per telefon besvara frågor från andra servicetekniker som arbetar ute på fältet med

att uppdatera och åtgärda fel på telefonväxlar. Arbetsbördan skulle avsevärt kunna minskas om serviceteknikerna på fältet kunde besvara sina egna frågor genom att själva söka information i handboken. Men detta hindras av otympligheten hos handboken. Den består av ett stort antal A4-pärmar, som knappast går att ta med sig överallt. Däremot skulle en elektronisk handbok göra det möjligt för serviceteknikerna att alltid ha med sig den önskade informationen ute på fältet.

## Användarcentrerad systemutveckling

Inom projektet har vi utgått ifrån att en lyckad utformning av ett användargränssnitt bl a baserar sig på empiriska studier av en tilltänkt användargrupp. Om man vid upprepade tillfällen under utvecklingen av ett system tittar på hur systemet används av personer som tillhör målgruppen för systemet kan man få svar på viktiga frågor rörande utformningen av gränssnittet. Man kan t ex få reda på om sättet som systemet är avsett att användas på stämmer med hur det verkligen används och vilken typ av svårigheter en användare kan råka ut för. Gränssnittet kan sedan förändras efter de slutsatser man drar från användarstudien. Den här typen av cyklisk och användarcentrerad

systemutveckling har utgjort stommen för projektets arbetsgång.

Under den första fasen i vår utveckling av handboksprototypen diskuteras tidigare nämnda metod, som utgår ifrån att betrakta den elektroniska handboken som ett informationssystem. Det finns inte här utrymme för att gå igenom metoden annat än mycket övergripande. Ett grundläggande steg är begreppsmodellering. Först analyseras handboken och en begreppsmodell över vilken information som finns tillgänglig i handboken tas fram. Denna begreppsmodell används sedan för att stycka upp textinnehållet i handboken. Utifrån begreppsmodellen skapas en relationsdatabas där de olika textstyckena lagras. Därefter går man igenom texten och olika hyperlänkar läggs in. Metoden kan sägas kombinera databasteknik och hypermedia-teknik.

Det finns flera fördelar med denna ansats jämfört med vanliga hypermediasystem (se faktaruta). En fördel är att begreppsmodellering är ett effektivt sätt att skapa en struktur på handboken. En annan fördel är att i och med att handboken är uppbyggd enligt samma principer som andra informationssystem i organisationen ökar förutsättningarna för att integrera handboken i verksamhetens to-



analysfasen som en karta för att ge användaren en överblick av innehållet i handboken.

Begreppskartan bottenar i en diskussion om på vilket sätt läsaren av handboken ska kunna tillägna sig en överblick av textens struktur och innehåll. I många hypertextsystem försöker man uppnå detta genom att visa en karta över hur alla noder är kopplade till varandra - en karta över nod-länkstrukturen. Men problem uppstår så fort man hanterar en större textmängd. Kartan som skulle underlätta överskådligheten blir i sig oöverskådlig. Begreppskartan i vår hanboksprototyp antar därför en något annorlunda form. I stället för att direkt representera en struktur av varje enskild nod och länk, representerar en låda i begreppskartan en kategori av hypertextnoder. Dessutom är den interaktiv på ett sätt som är tänkt att vägleda läsaren i hans sökning efter den information han behöver. Kategoriseringen av noder utfördes genom att analysera de redan existerande handbokstexterna.

## Servicetekniker i användarstudie

Målet för den empiriska användarstudien var att studera hur vår handboksprototyp skulle användas av representativa användare när de löste verklighetsanknutna uppgifter. En expert på materialet för vår handbok bistod med nio uppgifter som gick ut på att skriva kommandon för att konfigurera olika funktioner i telefonväxeln. Sex servicetekniker deltog som försökspersoner i studien. Av dessa sex hade två personer större kunskaper om innehållet i handboken än de övriga fyra. Dessa två arbetar dagligen med just den telefonväxel som handbokstexten handlar om medan de övriga fyra arbetar med en annan växel. Uppgifterna var dock så pass svåra att även personerna med kun-

skap om innehållet i handboken för varje uppgift behövde söka efter information.

Alla försökspersoner testades individuellt och under försökets gång loggades deras interaktion med prototypen. Innan de för första gången satte sig framför datorskärmen frågade vi hur de tolkade några av de begrepp som fanns i begreppskartan - "tjänst", "funktion", "kommando", "parameter", "maskinvara", "aktivitet" och "arbetsgång". De ombads också att lägga upp sina egna begreppskartor. Försöket avslutades efter två timmar.

## Olika söksätt

Alla försökspersoner klarade av att utföra de (eller del av) uppgifter de hann med under försökets begränsade tid. En person löste 3 uppgifter, två personer 2 1/2 uppgifter, en person 2 uppgifter, en person 1 1/2 uppgift och en person 1 uppgift. De två personerna med kunskap om innehållet i handboken löste 3 respektive 2 1/2 uppgifter. En analys av logutskrifterna visar att de två personer som hade kunskap om innehållet i handboken i högre grad än de utan sådan kunskap utnyttjade begreppskartan i sina sökningar. Medan begreppskartan stod för drygt 40% av alla sökningar hos den förra gruppen stod den för endast drygt 10% av sökningarna hos den senare gruppen. Vi vill här passa på att varna för att övertolka dessa siffror. Antalet försökspersoner är för litet för att man ska kunna dra starka slutsatser om hur personernas kunskaper förhåller sig till deras användning av begreppskartan. Det faktum att de oftare använde begreppskartan behöver inte bero på deras större kunskap om innehållet. Vad man däremot kan utläsa av detta resultat är att det pekar på intressanta frågor som ytterligare bör undersökas. Kan det exempelvis vara så att begreppskartan stödjer läsaren i att

utforska texten och att någon som har kunskap om textens innehåll känner sig tryggare och därmed är mer benägen att ge sig ut på utforskande vandringar i texten?

*"Den största enskilda förbättring man kan införa i en organisation är ett hypertextsystem där allas erfarenheter kan samlas."*

*Douglas Engelbart*

Vad beträffar de övriga sätten att söka i handboken visade det sig att användargruppen totalt ägnade sig mest åt att följa hypertext-länkar, d v s att klicka på nyckelbegrepp i texten för att komma till andra textavsnitt. Men historielistan var ett nästan lika populärt sätt att vandra i handboken. Här kunde man alltså välja att gå tillbaka till en viss plats i texten där man befunnit sig vid ett tidigare tillfälle. Man kanske vet att man tidigare sett text som kan vara användbar för det man ska göra just nu.

Totalt utnyttjades indexet i mindre utsträckning än de övriga söksätten. Men en uppdelning av indexanvändning med avseende på kunskap om innehållet i handboken visar att de försökspersoner som inte hade denna kunskap utnyttjade nyckelordsklickningar, historielistan och indexet ungefär lika mycket. Däremot använde de övriga två försökspersonerna indexet relativt lite. Personerna utan kunskap om innehållet använde indexet i 28% av alla sökningar medan de som kände till innehållet endast använde det till 8%.

Det finns alltså en skillnad på i vilken utsträckning försökspersonerna utnyttjade de olika söksätten. Det mest slående resultatet var att begreppskartan inte användes i större utsträckning av de försökspersoner som saknade kunskap om den telefonväxel



# Nytt från Kiseldalen

Från SISUs utsände vid Sveriges Tekniska Attachéer i San Francisco.

## Hej SISU-medlem (och ni andra också)!

Denna ruta kommer att bli ett återkommande inslag i ett antal nummer av Informa. Upprinnelsen är ett nyligen initierat samarbete mellan SISU och Sveriges Tekniska Attachéer i San Francisco.

I denna första ruta tecknas bakgrund och idé för samarbetet. Samtidigt passar vi på att göra lite reklam för verksamheten. De kommande rutorna kommer att innehålla ett antal korta notiser från Silicon Valley kring aktuella lanseringar, händelser, rykten m m.

Inom San Francisco-regionen (Silicon Valley) ligger den förmodligen största koncentrationen av framgångsrika företag för utveckling och produktion av såväl hårdvara som mjukvara inom Informationsteknologiområdet. Med Stanford- och Berkeley-universitetet som motorer bedrivs också det spektrum av avancerad forskning som är en nödvändig förutsättning för att området ska kunna vidmakthålla och förstärka sin ledande position.

Verksamheten vid Attachékontoret är inriktat mot att följa och rapportera om denna utveckling genom State-of-the-art rapportering, trend-bedömningar, produktanalyser, notiser m m. Kontoret hjälper också till att initiera kontakter mellan svenska och amerikanska företag/organisationer, lägga upp besöksprogram o s v.

SISU har som ett viktigt verksamhetsområde att för sina medlemmars räkning bedöma, sammanställa och rapportera om trender, erfarenheter, forskningsresultat o dyl inom SISUs kompetensområde. Att en del av denna insats fokuseras mot Silicon Valley är helt naturligt.

SISU och Attachékontoret har också klart överlappande, men framförallt, kompletterande målsättningar. SISU bevakar i huvudsak sitt kompetensområde i form av detaljstudier och fördjupande analyser. Attachékontoret har ett betydligt bredare bevakningsområde, innefattande både hårdvara, mjukvara, kommunikation och metoder. En naturlig följd blir att kontoret företrädesvis producerar och förmedlar översikter och kortare sammanställningar. När så anses lämpligt engagerar kontoret temporärt s k specialister för insatser inom ett visst, avgränsat område.

Just i denna roll fanns den naturliga, konkreta kontaktytan för ett samarbete mellan Attachékontoret, som etablerad organisation på plats samt med upparbetade kontaktytor, och SISU med sin specifika kompetensprofil. Ett samarbetsavtal upprättades med tidshorisonten 1 år. Inom avtalet avser SISU att lägga ut en viss volym uppdrag på Attachékontoret att i första hand utföras av mig som korttidsanställd (och samtidigt tjänstledig från SISU).

Sedan starten i slutet av februari 1992 har det SISU-relaterade arbetet bestått av uppdrag härstammande från TRIAD- respektive STANLI-projekten. Fortsatta aktiviteter inom dessa projekt är beslutade eller planerade.

Dock! Det finns till hösten utrymme för fler spännande uppdrag. Dels finns det luckor i min kalender, dels - inte att förglömma - finns vid attachékontoret ett antal duktiga och engagerade personer, var och en med sin specialinriktning.

Nedan följer, utan någon som helst rangordning, förslag på ett antal intressanta områden, inom SISUs intressesfär, som vi kan bevakna och förmedla i form av rapporter, newsletters, seminarier eller på annat sätt.

- Repositories (leverantörer, funktionalitet, användning, erfarenheter, trender)
- Objektorientering (metoder, OODBMS, erfarenheter, trender, aktuell debatt)
- Case-verktyg (leverantörer, funktionalitet, användning, erfarenheter, trender)
- 4G-verktyg (leverantörer, funktionalitet, användning, erfarenheter, trender)
- DBMS (leverantörer, funktionalitet, användning, erfarenheter, trender)
- Dataadministration (metoder, erfarenheter, datorstöd)
- Standardiseringsaktiviteter inom IT-området. ANSI och ISO. (Data Management, IRDS, ramverk, m m)
- Geografiska Informationssystem (modeller, produkter, standardiseringsaktiviteter i USA)
- Utvecklingsplattformar (standardisering, ramverk, produkter. Exempelvis CDIF, PCTE, AC/Cycle, Softbench, Cohesion, ....)
- Konferens- och mässbevakningar

Uppdrag från SISU-medlemmar förmedlas normalt genom SISU. För mer information, diskussion kring tänkbara uppdrag, vänd dig till någon av oss:

**Lars Bergman, SISU, Box 1250, 164 28 Kista.**

**Tel: 08-752 16 13.**

**Michael Gunnarsson (Teknisk attaché i San Francisco), Swedish Attaché Office, 120 Montgomery St., Suite 2165, San Francisco, CA 94104, USA. Tel: 0091-415 9827201. Fax: 0091-415 9827362.**

**Stig Berild, samma adress och telefonnummer som Michael.**

På återhörande!  
hälsar  
Stig Berild





Objektorienteringen får ibland en nästan religiös betydelse och allmängiltighet, med sina fanatiskt trogna tillbedjare. Dags för bot och bättring? Foto: Örjan Odelbo/Graffiti.

rimligt korrekt information om en annan verklighet. Datamodellen är en avspiegling av denna verklighet. Vi får finna oss i att registrera vad som händer där i form av aktiviteter, händelser, tillstånd, mm. Gärna med stöd av fina hjälpmedel som bankomater. Men vi måste hålla isär de objekt som är databärare (data) och de som är databehandlare (funktion).

– *Det visar bara att du är fastlåst i din gamla föreställningsvärld.*

### **Datamodellering - objektklassmodellering**

– Jag trodde faktiskt att jag var lite grann objektorienterad eftersom jag brukar koncentrera mycket utvecklingsarbete på att tidigt finna dom objektklasser och samband som behövs för att ta del av och för att hantera information. Baktanken är att objekten består (är stabila) men funktioner förgår (är flyktigare, mer i behov av löpande anpassning mot en föränderlig verksamhet).

– *Men du arbetar bara med de statiska objekten.*

– Javisst. Inom datamodelleringsområdet finns en tjuogoårig tradition (Abrial, Senko, Bubenko, Chen, m fl) där du hittar en semantisk uttryckskraft, som OO-området nu steg för steg börjar upptäcka och göra till sin egen. Hur stämmer förresten inkapslingsidén inom OO med den rika förekomsten av fixa samband som vanliga datamodeller arbetar med?





loken, låntagaren och bibliotekarien vara separata objekt? Foto: Susanne Sandström/Graffiti.

– Kan så vara, rent teoretiskt. Det finns regler för hur länge böcker får lånas. Men dessa regler ligger ju inte naturligt i lånens intresse att bevaka. Det berör strategier och regler i biblioteksrollen som helhet. Att de ändå tenderar hamna hos låneobjektet kan ju bero på att detta är det närmast berörda. Regeln finns i ett exemplar men berör var och en av ett stort antal objekt. Låneobjekten och det som opererar på låneobjekt bör skiljas åt. Vid vårt bibliotek där hemma vill man förresten skicka ut *en* påminnelse för samtliga böcker som en person borde ha återlämnat. Den rutinen brukar bibliotekarien, eller någon annan anställd också för den delen, initiera när han eller hon tycker att det är lång väntelista på en del efterfrågade böcker. Helt enkelt en stödfunktion till förlo-

– Jaha.

### Funktion som funktionsobjekt

Påtagligt uppe i varv fortsätter Arne:

– Stödfunktioner kan ibland naturligt vara fokuserade på ett visst objekt, i andra fall på flera objekt. Stödet ges i form av ett antal villkorliga och/eller ovillkorliga steg, som tillsammans bildar en helhet (dialogstruktur, rutin-skiss...) till någon avgränsad del av en verksamhet. Att 'registrera lån' och att 'uppdatera ett bankkonto' är operationer som hjälper en iakttagare att förmedla informationen kring en händelse, förmedla ett nytt tillstånd. Man tar en kamerabild på verkligheten.

– Det är inte objektens beteende vi ska modellera utan hur och när de ska fotograferas. (Däremot kan det ju vara alldeles utmärkt att skapa kameran som en uppsättning samverkande, dynamiska objekt.) Ibland fotografe-

rar vi ett enskilda objekt. De olika lämpliga motivvinklarna för detta kan ju samlas under en rubrik som är objektets namn. Består motivet av flera objekt får man hitta någon annan benämning som kanske mer har bäring på motivet som helhet än de ingående objekten. Inom OO tycks man utgå från att dataflöde, beteende m m alltid kan stoppas in i ett objekt. Visst går det med ett antal nödgrepp, men vad har vi kvar av perspektivet på naturlig *avbildning av verkligheten* i en OO modell?

– Jag blir så trött.

Bakåtsträvaren tar in ny luft:

– För övrigt har funktionsobjekten i administrativa sammanhang sällan några komplicerade beteenden och tillstånd, utan bara olika värden på intressanta egenskaper. De funktioner som behövs (lagra, radera, uppdatera, visa) återfinns färdigutvecklade enligt standardiserade (!) gränssnitt hos databashanterare. Även formulärhanterare finns att köpa.

### Objekt och objektklasser

I ett försök att mjuka upp den nu ansträngda atmosfären försöker Arne:

– OO-traditionen härstammar från en tillämpningsvärld där man inte har någon uppsjö objekt att handskas med. Varje objektkonstruktion innebär ett hantverksarbete. I det läget kanske inte den här uppdelningen mellan data och funktion spelar så stor roll? Man har inte ens alltid känt behov av att dela upp i objektklasser och objekt-förekomster, något som är en av hörnpelarna vid modellering i administrativa sammanhang.

– Administrativa funktioner blir också ofta generella och permanenta med uppgift att om och om igen operera på en eller flera objekt-förekomster ur specifika objektklasser. Jämför med biltvätt (funktion) och bilar(objekt). Det blir lättare att skilja på funktion och data då.



# STORT INTRESSE FÖR LATURI

**D**et finns ett stort behov av att bättre kunna beräkna tid och kostnader för att utveckla nya informationssystem. SISU kan erbjuda ett stöd för detta – Laturi. Det är framtaget av TIEKE, SISUs systerinstitut i Finland. Där har Laturi blivit mäktigt populärt och efterfrågan bara ökar.

Den 15 april 1992 hölls det första seminariet för att presentera den svenska versionen av Laturi. Salen var fylld till bristningsgränsen. Trots det kunde inte SISU ta emot alla intresserade.

Som läget är nu slår estimeringarna fel på i genomsnitt 30 procent. Detta är en ohållbar siffra, vi måste helt enkelt bli bättre på att räkna. Som en liten jämförelse kan nämnas att om byggbranschen gjort en felbedömning på 6 procent, anser de sig ha räknat dåligt. Vad är det som orsakar de stora felbedömningarna?

Det är många faktorer som spelar in här, men det kan t ex bero på:

- Dålig planering och styrning av projektledarna.
- Att uppdragsgivaren vill ha mer. Under projektets gång stiger ambitionerna.
- De estimeringsansvariga underskattar värdet av kunskap. Personalen som går in i projektet är inte tillräckligt kunnig.

- Effekten av ny teknik är kanske okänd.
- Projektdeltagare underskattar komplexiteten, systemet de ska utveckla är svårare än de trodde.
- Man räknar på för dåliga underlag.

För att minimera felbedömningarna ställs det höga krav på estimeringen. En av föreläsarna, Christina von Greyerz från Programator, säger att hon inte hittat någon metod som kan hantera alla dessa problemspekter. Enda sättet har varit att kombinera olika metoder. Det är också till stor hjälp att jämföra olika projekt, men då är det svårt att veta vilka projekt som är jämförbara.

Inte ens Laturi kan lösa alla dessa problem, men Christina har inte hittat något verktyg som klarar av det bättre än Laturi.

## Bygger på funktionspoängs- metoden

Laturi är en finsk förkortning för Kvalitet, Produktivitet och Riskanalys.

Det är ett verktyg för:

- Planering av ADB-projekt
- Kostnads- och tidsuppskattning
- Produktivitetsuppskattning och -mätning

Vem har användning av Laturi?

- Projektledare
- Estimeringsansvariga
- ADB-chefer
- Beställare/Kund

För en bra estimering krävs ett lätt-använt verktyg som:

- Har samma beräkningsmetod för varje projekt.
- Ser på informationssystem på ett enhetligt sätt i alla projekt.
- Har speciellt framtagna beräkningar/formler. Det ligger lång forskning bakom dessa.
- Utnyttjar andras erfarenheter.

Allt detta är mycket krångligt att komma fram till, därför har verktyget Laturi kommit till. Kraven för god estimering finns inbakade i Laturi.

För att göra en bra estimering behöver man veta systemets storlek. För detta använder man sig av funktionspoängmetoden, (se bild).

Hur går man tillväga?



## SÅ TYCKTE DELTAGARNA

Vi frågade några av deltagarna på seminariet hur de uppfattat estimeringsverktyget Laturi. Här är frågorna och svaren.

### Vad fick du för intryck av Laturi?

**Ingmari Hellström, Enea Data AB:**

– Det behövs något i den stilen. Det finns ett behov. Laturi ger bra information och det är bra med ett sånt hjälpmedel.

**BrittMarie Bjurström, Arbetsmarknadsstyrelsen:**

– Vi har behov av en sån produkt. Jag fick ett bra intryck.

**Lars Abrahamsson, S-E-Banken Data:** – Rätt så positivt. Hur Laturi hanterar informationen var bra.

**Ragnar Tofvesson, WASA ADB, Stockholmsenheten:**

– Oftast då man gör en utvärdering har man några liknande produkter att jämföra emellan. Men vad gäller Laturi finns ingen motsvarighet. Jag tycker att det är en positiv produkt och mycket användbar. Den är nog efterlängtd av många företag. Helt utvecklad är den inte, man skulle kunna få den mer precis.

### Tror du att Laturi kan vara något för dig?

**Ingmari Hellström, Enea Data AB:**

– Inte för mig personligen. För företaget skulle det kunna vara något, men då beror det på priset. Priset är viktigt när det gäller ett litet företag.

**BrittMarie Bjurström, Arbetsmarknadsstyrelsen:**

– Det skulle kunna vara något för oss, vi har behov av det. Men det är lite tidigt att säga just nu.

**Lars Abrahamsson, S-E-Banken Data:** – Ja, det tror jag. Det skulle kunna vara mycket intressant. Men problemet kan vara att få det att slå igenom i en sån här stor organisation. Implementeringen i företaget kan bli svår. Att få dem som ska använda Laturi att göra det på ett riktigt sätt.

**Ragnar Tofvesson, WASA ADB, Stockholmsenheten:**

– Ja, det tror jag. Jag arbetar på systemteknik och där arbetar vi bl a fram säkerhet, effektivitet och kvalitet. Det är precis vad Laturi vill ha också. Huvudmålet för alla borde vara att ge precisa resursestimeringar på kundens krav.

### Tycker du att något var speciellt bra eller speciellt dåligt med Laturi?

**Ingmari Hellström, Enea Data AB:**

– Databasen var bra. Idén om hur alla användare levererar erfarenhetsdata från egna avslutade projekt är också bra.

**BrittMarie Bjurström, Arbetsmarknadsstyrelsen:**

– Jag tyckte om Laturis struktur. Alla produktivetsfaktorer som togs upp var bra. Det var bra punkter att ta fasta på, man känner igen sig.

**Lars Abrahamsson, S-E-Banken Data:** – Den interaktiva delen var bra. På det sätt man liksom höll en dialog med verktyget. Det var bra att man kunde ändra olika indata och på det viset se ett annorlunda utslag. Verktyget var lätt att arbeta med, automatiken var bra.

**Ragnar Tofvesson, WASA ADB, Stockholmsenheten:**

– Automatiken var bra. Man kan lägga in olika värden. Laturi gör jobbet.

När man utvecklar ett system kostar det alltid mer att hålla det vid liv än att skapa det. Laturi borde satsa mer på 20/80-regeln. Kostnaden för skapandet av systemet ligger på 20 procent men sedan ligger den verkliga kostnaden, 80 procent, på förvaltandet. Det finns inget gångbart system i Laturi att estimerar denna del av kostnaderna, de kostnader som är den riktiga tyngdpunkten. Laturi borde i alla fall så när som till 70 procent uppfylla förvaltningsskedena.



FRAMTIDENS INFORMATIONSNÄTVERK:  
**SVERIGE SACKAR I  
UTVECKLINGEN**

**N**ästa generations informationssystem omspänner stora områden. Från informell modellering av verksamheter till tekniska, formella problem som rör till exempel synkronisering av långlivade transaktionsprocesser i decentraliserade system. Detta ställer stora krav på kompetens och kunskapsöverföring. Krav vi idag inte lever upp till i Sverige.

Under två workshops (Niagara-on-the-Lake, Ontario och Como, Italien) har ett trettiotal "visionära" dataproffs och forskare från USA, Kanada och Europa diskuterat framtidens, eller Nästa generations informationssystem.

Deltagarna är så valda att ett flertal delområden inom datavetenskap blir representerade: programmeringsspråk och metodik (PL), programvaruteknik (software engineering (SE)), informationssystem (IS), databasteknik (DB), artificiell intelligens (AI), och män-

niska-dator-interaktion (HCI), och datorstött (sam)arbete (CSCW), och kontorsinformationssystem (OIS), m fl. Härigenom vill man uppnå en allsidig belysning av frågeställningarna och en diskussion som spände över de kända datavetenskapliga "subkulturerna" och paradigmen.

En tredje workshop i Dagstuhl, Tyskland, med temat "Distributed Cooperation in Integrated Information Systems", kommer Janis Bubenko att rapportera ifrån senare.

Deltagarna i den internationella arbetsgruppen ombads att kommentera och ge bidrag till följande scenario om framtidens system.

Nästa generations informationssystem kommer att utmärkas av ett stort antal intelligenta agenter (datorer, människor) fördelade över ett stort, kraftfullt kommunikationsnät. Ett starkt decentraliserat arbetssätt kommer att gälla. De olika delsystemen i ett sådant nät kommer att utmärkas av en hög grad av autonomi. Det kommer inte att finnas någon global kontroll över de enskilda nodernas resurser eller deras begreppsapparater. Heterogenitet på det tekniska planet såväl som på det tillämpningsorienterade planet (den semantiska nivån) kommer att gälla. Dessa systems arbetssätt

kommer att utmärkas av bi- eller multilaterala förhandlingar och kontrakt om resursdelning, samarbete, import och/eller export av resurser och tjänster, och gemensam problemlösning.

#### **Niagara-on-the-Lake**

Konferensens fokus var inställt på **begrepp, metoder och verktyg för utveckling av intelligenta och samverkande informationssystem** enligt scenariot ovan. En sammanfattning av konferensens diskussioner är publicerad [Huff, K. and J. Mylopoulos, "International Workshop on the Development of Intelligent Information Systems", Dept. of Computer Science, Univ. of Toronto, Canada, April, 1991]. Samtliga "position papers" finns redovisade i [Balzer, R. and J. Mylopoulos, International Workshop on the Development of Intelligent Information Systems", Niagara-on-the-Lake, Ontario, 1991, Dept. of Computer Science, Univ. of Toronto]. Här redovisar jag några personliga reflektioner.

- Gapet mellan användarnas uppfattning om det system de behöver och det som dataexperterna specificerat ansågs som det svåraste och kanske viktigaste problemet. Hur kontrollera validiteten, dvs att en kravspecifikation





Ett nätverk av forskare diskuterade, på två konferenser, nätverk av intelligenta, samverkande informations-system. Foto: Kenneth Hellman/Graffiti.

förstå varandra. Viljan tycks finnas men "paradigmerna", d v s uppfattningen om vilka som är de viktiga forskningsproblemen och hur dessa ska angripas, skiljer sig mer eller mindre mellan deldisciplinerna. Det föreföll dock som om flertalet var överens

om vilka som var de svåra respektive mindre svåra problemen. Till de mycket svåra problemen räknades framför allt att avgöra huruvida en systemspecifikation motsvarar användarnas och tillämpningens krav, d v s valideringsproblemet.

(Politecnico, Milano) konstaterade inledningsvis det stora behovet i näringslivet av att hantera evolution och interoperabilitet hos befintliga, separat utvecklade system och databaser. Det stora problemet är således att hantera arvet av gamla och funktionellt otillräckliga system.

### COMO-konferensen

Konferensens fokus var inställt på grundläggande teknologier för Nästa Generations Informationssystem (NGIS), kommunikation och samverkan (interoperabilitet), stöd för hantering av stora, decentraliserade data-, informations-, objekt-, och kunskapsmängder, intelligenta system [Brodie, M.L and S. Ceri, "Proceedings of the Second International Workshop on Intelligent and Cooperative Information Systems: Core Technology For Next Generation Information Systems", 1991, Villa Olmo, Como, Italy]. Man definierade tre nivåer av samverkan.

1. Kommunicerande system med autonoma delsystem (noder) som kan utväxla/kommunicera meddelanden
  2. Interoperabla system, där autonoma delsystem kan exportera och importera tjänster av olika slag
  3. Samarbetande (cooperative) system, där autonoma delsystem samverkar för att lösa problem.
- Como-workshopen blev mest inriktad på system på nivå 2. Brodie (GTE Labs, USA) och Ceri



ditionella datavetenskapliga, och ofta teknologiskt snäva, synsättet. De mest aktuella trenderna just nu är att skapa datorbaserade modeller av verksamheter som sedan nyttjas för olika analyser och verksamhetsstödjande processer. I dessa sammanhang spelar ofta de politiska, organisatoriska, sociala, och psykologiska aspekterna en mer avgörande roll än de rent teknologiska.

### Modellering och Repository

Atzeni och Batini (Univ. di Roma "La Sapienza") talade om det växande behovet av "verklighetsmodellering" som ett led i att bygga framtida, kunskapsbaserade system och "knowledge dictionaries". Det finns behov av ökad kunskap om transformation av modeller mellan olika representationer och om metoder att stegvis bygga och integrera modeller av tillämpningar (verkligheter). Även Mylopoulos (Univ. of Toronto) anslöt till modellerings- och repositorytemat. Han ansåg att "Conceptual Modelling is a core technology for future ICIS (Intelligent and Cooperative Information Systems)" och pekade på vikten att skilja på modellering av verklighet respektive modellering av formella informationssystem. Båda typerna är väsentliga komponenter i en systemutvecklingsprocess.

### Samverkan och kommunikation

Bubenko beskrev SISUs arbete med att utveckla en teknik för samverkan i sk federerade system (inom ett ESPRIT-projekt). De problem som man främst tacklar är kommunikation mellan autonoma noder, import och export av information och meta-information, schema- och dataintegration och hantering av semantiskt heterogena miljöer. Bubenko ansåg att hantering av semantiska tolknings- och

översättningsproblem vid informationsutbyte tillhör ett av de viktigaste framtida forskningsproblemen.

Balzer (USC ICS, USA) och Reuter (Univ. of Stuttgart) diskuterade olika operativsystem- och programmeringsmekanismer för hantering av samverkan och kommunikation i decentraliserade, autonoma system. Ett viktigt problem här är att hantera, styra och synkronisera sk "långlivade, distribuerade aktiviteter" (transaktioner, "scripts") i system av autonoma noder, där en sådan komplex, långlivad process kräver resurser/service från olika noder. Här finns en rad forskningsproblem som bl a rör tillförlitlighet, konsistens, konflikthantering, m m.

Garcia-Molina (Princeton) ansåg att den nu kända och tillämpade tekniken för styrning av "concurrency control and recovery" i distribuerade system inte kommer att vara tillräcklig i framtida system. Dagens mekanismer förutsätter i alltför hög grad att kopplingar mellan system är tillförlitliga och att data alltid är tillgängliga. Detta kommer inte att gälla för framtidens autonoma, mer öppna, löst kopplade system. Han diskuterade därefter hantering av sk "application constraints" och hantering av dataosäkerhet i decentraliserade system.

### Evolution till framtida system

Stonebraker (Berkeley) diskuterade problemet "vad gör man med organisationer som underhåller 10 år gamla eller äldre system, bestående av miljontals (COBOL) instruktioner, men som utför affärsmässigt kritiska uppgifter?". Systemen är praktiskt taget omöjliga att förändra och vidareutveckla. Vilken strategi bör man förordas i ett sådant fall? Att skriva om hela system i modern programvara? Ett sådant alternativ innebär alltid ett stort risktagande och kräver dessutom

lång utvecklingstid och stora, oftast okontrollerbara, resursinsatser. Stonebraker förordade ett inkrementellt arbetssätt - d v s att arbeta om systemet i små, överblickbara (mindre än 10 personår) steg. Men ett sådant arbetssätt kräver att det nuvarande systemets struktur och funktionalitet kartläggs väl och att dess funktionella kärna tydligt identifieras. Men man behöver också arbeta på att utveckla bättre sk "surround technology" (teknologi som bakar in gamla system och ger dem ett mer flexibelt och kraftfullt gränssnitt).

*If a train-station is a place where trains stop...  
and a bus-station is a place where buses stop...  
then what is a work-station?*

Kanske det viktigaste var att arbeta på "specification extractors" - d v s "reverse engineering" teknik. Enligt Rosen (CityCorp) var konceptuell modellering ett effektivt hjälpmedel dels för att förstå affärsverksamhetens kärna dels för att isolera och beskriva informationssystemens funktionalitet.

Är det här typen av problem någonting som den moderna datavetenskapliga forskningen ska bry sig om? Vi ska kanske utveckla den frågan vid ett annat tillfälle, men det bör redan här påpekas att högskolebaserad forskning som direkt eller indirekt angriper det sk karvsproblemet (att hantera arvet av många gamla, nerkörda och ofta funktionellt bristfälliga system) inte har givits den status och den prioritet som motsvarar problemets samhällsliga och ekonomiska betydelse.

Det framhölls på workshoppen att en del (teoretiker) anser att arvsproblemet är vetenskapligt "orent", rentav "fult". Dessa hoppas på att problemet på något vis ska lösas eller försvinna. Omvändningen kommer



# L SATSNING PÅ FORSKNING INFORMATIONSSYSTEM

En icke ringa del av seminariet ägnades åt diskussioner om lämpliga åtgärder för ökad samverkan.

Resultatet av seminariet blev förslag till satsningar i form av två ramprogram:

1. Ett nationellt program för forskarutbildning i informationssystem.

Syftet med detta program är att utnyttja, stärka och bredda den nationella basen av kvalificerade forskare i informationssystem såväl vid högskolorna som i näringslivet.

2. Ett program för tillämpad forskning om Informationssystem.

Programmet ska främja sådan behovsbaserad (tillämpad) forskning i informationssystem, som på 5 års sikt kan omsättas praktiskt och bidra till att öka det svenska näringslivets affärsutveckling och konkurrensförmåga. Detta ska ske genom förbättrad användning av informationsteknologi och effektivare informationshantering, kvalitativt såväl som produktivitetsmässigt.

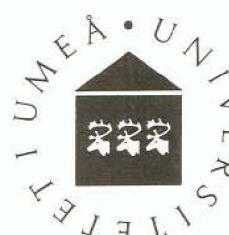
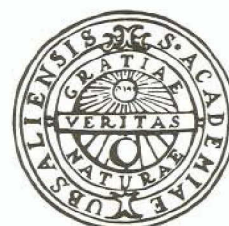
Det utarbetade förslaget ger först en syn på behovet av ökad forskning i området. Därefter presenteras kortfattat förslag till de två programmen och åtgärder för att mer i detalj utforma dem.

Deltagarna har lämnat underlag, bidrag till och synpunkter på innehållet till de två ramprogrammen. Förslaget har överlämnats till NUTEK.

Av Janis Bubenko

I seminariet deltog:

Prof. Janis Bubenko, SISU  
Tf. Prof. Bo Dahlbom, CTH/GU  
Prof. Pelle Ehn, LU  
Prof. Thomas Falk, CTH och SISU  
Doc. Göran Goldkuhl, LiU  
Prof. Åke Hansson, UU  
Prof. Sture Hägglund, LiU  
Prof. Bengt Lundberg, SU  
Dr. Anders Nilsson, HHS  
Dr. Björn Nilsson, SISU  
Tf. Prof. Kenneth Nilsson, UmU  
Prof. H-E Nissen, LU  
Prof. Jacob Palme SU  
Prof. Bo Sundgren HHS.





# RAPPORT SOM RÖJER VÄG I GRÄNSSNITTSDJUNGELN

**I** SISU Rapport nr 20 görs en lång vandring i gränssnittsjungeln. Den som läser rapporten kommer att gå betydligt säkrare i fortsättningen. Marknadens utvecklingshjälpmedel för konstruktion av användargränssnitt beskrivs nämligen utförligt och jämförs sedan med varandra - rapporten är som en machete för den som inte vill fastna i den snabba utvecklingens snåriga växtlighet.

Ju mer lättanvänt ett gränssnitt (och därmed ett program) är, desto svårare är det att konstruera.

Det är lätt att förstå vikten av att program är lätta att använda. Därmed förstår man också hur viktigt det är att den som ska konstruera gränssnittet har bra verktyg till sin hjälp.

Problemet för den som ska välja är att det finns en uppsjö av sådana verktyg och att de är bra på helt olika saker och följer helt olika standarder (nämligen de som beskrivs i SISU Rapport nr 19).

## Nya krav på programmerarna

En sak att ha i bakhuvudet är att det inte finns några riktlinjer eller standarder som *garanterar* att programmen blir lättanvända och användbara.

För att de ska bli det krävs mycket av en bred grupp programmerare: att ständigt - och snabbt - förbättra sina prototyper och att alltid hålla koll på nya konstruktionsverktyg som förenklar kodningen av gränssnittet.

Rapporten *Verktyg för grafiska användargränssnitt* kan då vara till god hjälp. Peeter Kool, Ulf Wingstedt och Per Bergsten har tillsammans gått igenom grunderna för olika typer av gränssnittsverktyg, jämfört de olika verktyg som finns på marknaden idag och pekat ut några viktiga utvecklingstrender.

Fjorton verktyg har utvärderats: *Case:W*, *Craftman*, *Devguide*, *Easel*, *Eyescream*, *EZX*, *Hypercard*, *Interface Builder*, *Open Interface*, *Plus*, *Telease*, *Visual Basic*, *VUIT* och *XVT*. Med hjälp av vart och ett av verktygen har rapportförfattarna byggt två sorters prototyper för att konkret kunna jämföra verktygens användbarhet.

## Visuella gränssnittsverktyg

Alla de utvärderade verktygen är så kallade visuella gränssnittsverktyg. Med det menas att utvecklaren redan när han konstruerar gränssnittet kan se det mycket klart på skärmen.

Verktygen bedöms efter hur avancerade de är (layout-, prototyp- eller totalverktyg). De delas in i grupper bland annat beroende på i vilka grafiska miljöer de fungerar, ifall de separerar gränssnitt och tillämpning, ifall de är objektorienterade och beroende på vilket programmeringsspråk utvecklaren måste använda.

En slutsats är att verktygen skiljer sig åt väldigt mycket. Vilket verktyg som passar bäst beror naturligtvis på i vilken miljö och till vad det ska användas.

Rent allmänt kan man dock säga att de idag kraftfullaste verktygen bygger på ett objektorienterat kodbibliotek. Den som vill veta mer hänvisas till rapporten.

Av Pål Török



## SISU-publikationer

SISU Analys och SISU Rapport är skriftserier som tas fram inom SISUs ramprogram och finansieras av intressenterna. SISU-skrifterna säljs till intressenterna och till högskolor och andra organisationer av forskningskaraktär till självkostnadspris och till icke intressenter till priser markerade med \*. Kontaktpersonerna får i regel två exemplar av SISU Analys och Rapport. Övriga upplysningar lämnas av Helena Granlöf, Marianne Hedberg eller Lars Bergman på SISU, tel 08-752 16 00.

### SISU rapport

- nr 1/87: Ett förslag till referensmodell för Människa-Dator-interaktion, 50 kr/150 kr\* exkl moms
- nr 2/87: Generellt teknisktöd för ärendehantering, 100 kr/300 kr\* exkl moms
- nr 3/88: En experimentell studie av CASE-verktygen Deft och IEW/WS, 140 kr/420 kr\* exkl moms
- nr 4/88: RAMATIC på Volvo Personvagnar, 80 kr/240 kr\* exkl moms
- nr 5/89: HYBRIS - A first step towards efficient information resource management, 80 kr/240 kr\* exkl moms
- nr 6/90: Konceptuell modellering med naturligt språk, 100 kr/300 kr\* exkl moms
- nr 7/90: Generering av naturligt språk från konceptuella scheman, 80 kr/240 kr\* exkl moms
- nr 8/90: DA Varför, Vad och Hur?, 350 kr/875 kr\* exkl moms
- nr 9/90: Kvalitet hos konceptuella scheman, 100 kr/300 kr\* exkl moms
- nr 10/90: VISION•95, Ett arbetsmaterial utarbetat av ISVI:s programkommitté, 150 kr/450 kr\* exkl moms
- nr 11/91: Reverse modeling from Relational Schemata to Entity-Relationship Schemata, 150 kr/450 kr\* exkl moms
- nr 12/91: En utvärdering av Hybris, 150 kr/450 kr\* exkl moms
- nr 13/91: Objektorientering- de vanligaste begreppen, 150 kr/450 kr\* exkl moms
- nr 14/91: Mot nya djärva språk, 150 kr/450 kr\* exkl moms
- nr 15/91: Business Engineering – en kritisk översikt, 150 kr/450 kr\* exkl moms
- nr 16/91: Modelleringensansatser för begrepps- och data-modellering, 150 kr/450 kr\* exkl moms
- nr 17/91: DVI - Digital Video Interactive – en teknikutvärdering, 100 kr/300 kr\* exkl moms
- nr 18/91: Överblicksproblemet i hypermedia, 100 kr/300 kr\* exkl moms

Namn \_\_\_\_\_

Befattning \_\_\_\_\_

Avdelning \_\_\_\_\_

Företag/organisation \_\_\_\_\_

Adress \_\_\_\_\_

Postnr och ort \_\_\_\_\_

Telefon \_\_\_\_\_

- nr 19/92: Standarder för grafiska gränssnitt, 150 kr/450 kr\* exkl moms
- nr 20/92: Verktyg för grafiska gränssnitt, 150 kr/450 kr\* exkl moms

### SISU analys

- nr 1: Konceptuell Modellering (1985)
- nr 2: Några aspekter på kontorsinformationssystem (1985)
- nr 3: Grafiskt baserade datorstöd för systembeskrivning (1986)
- nr 4: ADA-teknologi (1986)
- nr 5: Databaser - enkla att hantera (1987)
- nr 6: An Introduction to Distributed Database Systems (1987)
- nr 7: Kunskapssystem (1988)
- nr 8: OSI (1988)
- nr 9: Meddelandehanteringssystem (1988)

SISU Analys kostar 100 kr/300 kr\* exkl moms per nummer.

### SISU informa

- Sänd mig SISU informa fortlöpande (t.v. kostnadsfritt även för icke intressenter)

### SISU övrigt

- ex av CASE89-föredragen, ca 700 sidor, 675 kr exkl moms
- ex av förteckning över SISU-dokument, kostnadsfri
- ex av SISU:s kurskatalog, kostnadsfri
- ex av Erfarenheter från användning av Hybris, kostnadsfri