SISU informa

Nr 89/2

SISU har tagit modellering som medel för att driva utveckling av organisationen. Vid SISUs och ISVIs internat i april modellerade styrelserna SISUs mål och medel , problem och åtgärder.

Vid SISUs sedvanliga internkonferens i slutet av maj arbetade vi igenom en första vända av modeller för mål och verksamhet inom de områden SISU verkar inom. Bilden visar större delen av SISUs personal vid genomgång av KBS-områdets modell, som presenterades av Sten-Erik Öhlund.

INNEHÅLL

SISU informa utges av Svenska Institutet för Systemutveckling.
Ansvarig utgivare: Janis Bubenko jr, tel 752 16 00.
Redaktionen: Lars Bergman & Lottie Eriksson
Adress: Box 1250, 164 28 Kista. Besöksadr: Electrum, Kista. Tel. 08- 752 16 00. Fax: 08- 752 68 00

Inledaren

SISU accelererar

CASE89

Broslagning och perspektiv blev resultatet av den länge och mycket förberedda CASE89-konferensen. Ett av syftena för SISUs verksamhet är att slå broar mellan praktik och forskning. Därför var det glädjande att både föreläsare och deltagare representerade en god fördelning mellan praktiker och forskare. Totalt samlade CASE89 strax under 300 deltagande medräknat 50-talet föreläsare.

Perspektiv internationellt gavs genom att ungefär hälften av föreläsarna kom från utlandet. Ett tidsmässigt perspektiv fick man genom föredrag från dagens praktik via industrins och leverantörernas utveckling till forskningens mera långsiktiga verksamhet.

CAiSE90 - 8--10 maj 1990

The second Nordic Conference on Advanced information Systems Engineering, 8 - 10 maj 1990 i Electrum i Kista. Programkommitténs ordförande är professor Arne Sölvberg från Trondheims Tekniska Högskola och SINTEF.

IAS-89

Modellering var tema för årets IAS-konferens som rapporteras i kommande informa.

Publikationer

Två rapporter har just distribuerats till kontaktpersonerna. I **SISU rapport nr 3/88** redovisar **Tapani Kinnula och Jalal Matini "En experimentell studie av CASE-verktygen Deft och IEW/WS.** Tapanis föredrag på CASE89, som återges i detta nr av informa, grundar sig på den experimentella studien.

Stig Johansson då Volvo Personvagnar, nu SISU i Göteborg redovisar i **SISU rapport nr 4/88** erfarenheter från arbetet med **"RAMATIC på Volvo Personvagnar"** inom VDDS-projektets ram.

Missade du CASE89, så finns ett litet antal av konferensdokumentationen (ca 700 sidor) hos oss.

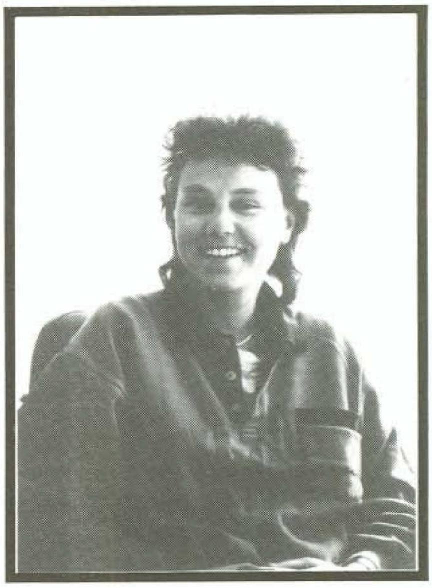
Behöver du förklara vad modellering är för någon t ex i utbildningssammanhang? - Vi har ett hjälpmedel för detta! **Broschyren "konceptuell modellering"** förklarar på ett mycket pedagogiskt sätt vad och varför. Den är gjord för att användas. Provexemplar är fritt.

Nya intressenter

Försvarets Materielverk med **Barbara Hedlund** samt **Vägverket** med **Björn Oresand** som kontaktpersoner, hälsar vi hjärtligt välkomna till ISVI- och SISU-samverkan!

Nya ansikten på SISU

Monika Korinek, på bilden, kommer från systemvetenskaplig linje i Uppsala och arbetar nu inom SISUs område KBS.



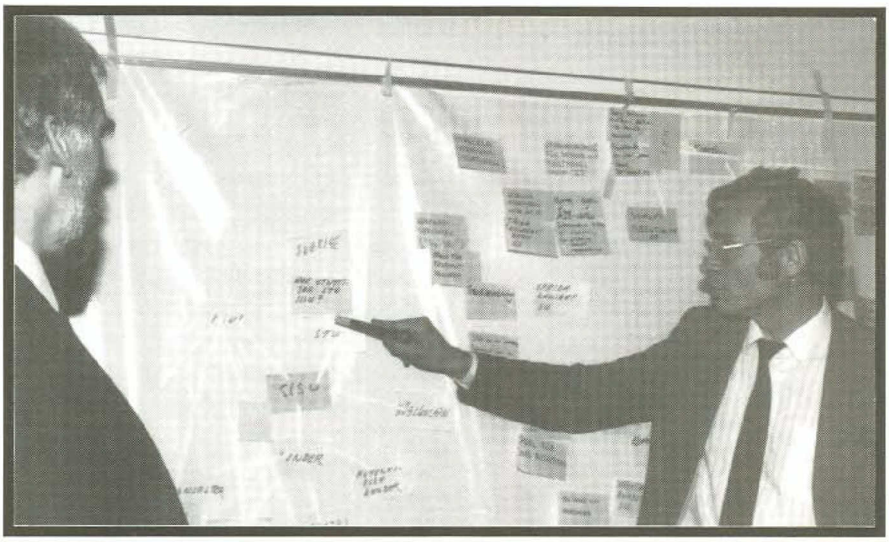
Clary Sundblad liksom **Marianne Jannings** samt **Stig Berild** kommer närmast från Infocon. Clary och Marianne arbetar inom SISUs område Modeller och metoder. Stig håller bl a i kompetensnätet Objektorienterad systemutveckling, KOS.

Ulf Persson, arbetar i Espritprojektet Tempora.

Jonas Olsson och **Per Bergsten** kommer från konsultverksamhet och förstärker SISUs område CASE i Göteborg.

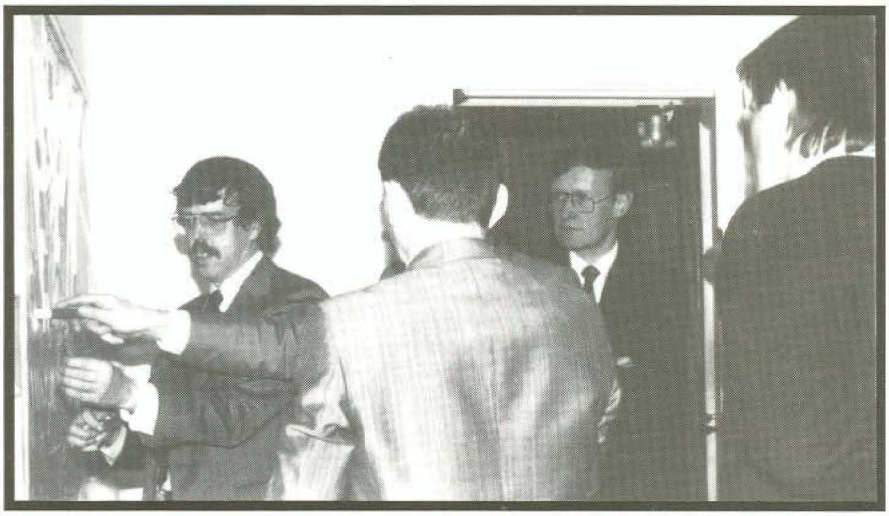
Styrelser modellerar SISU

Lars-Erik Dahlgren, SAF, pekar på frågan om hur STU skall utnyttja SISU för Björn Nilsson.



SISUs styrelse och ISVIs styrelse har gemensamt arbetat med SISUs mål och problem samt indikerat åtgärder och områden för åtgärder vid ett internat i april. SISUs personal har arbetat med SISUs mål, problem samt indikerat åtgärder och områden för åtgärder inom de olika områden institutet är verksamt. - En verkningsfull process har kommit igång med modellering som medel för processen, som ännu bara är inne i sitt första steg.

Bilden till höger: fr. v. Örjan Odelhög, Data Logic, Hans Holmberg, Telub Teknik (med ryggen mot kameran, John Fürstenbach, Stockholms universitet samt Lars-Åke Johansson, SISU i modelleringstagen.

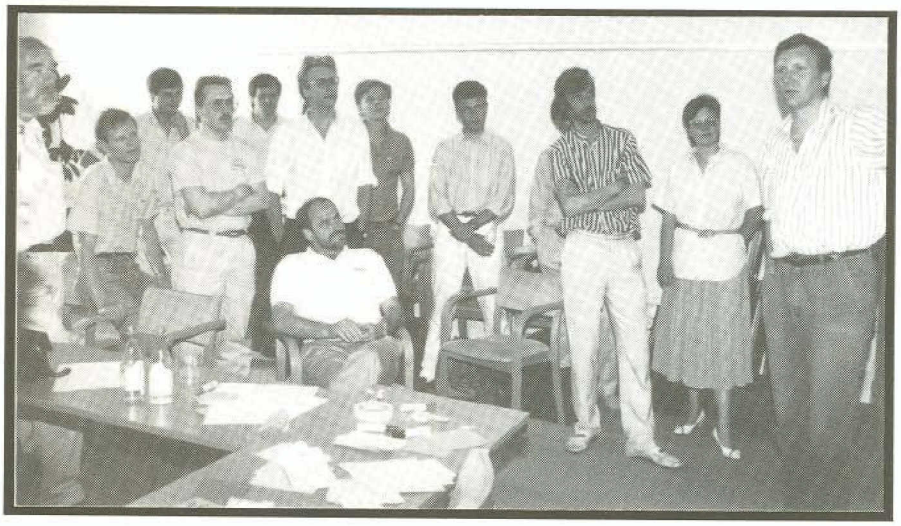


Staffan Westerberg, Digital och Erik Knudsen, SISU i lyssnartagen.

Henry Samuelsson, Televerket ADB-service och Christer Dahlgren, Ericsson/IT-plan.

Fr. v. Henry Samuelsson, Televerket ADB-service i modellering med Marianne Sindler, SISU, Bengt Carlefall, ADB-kontoret i Göteborg, Inge Dahlberg, Unisys och Barbro Atlestam, STU.

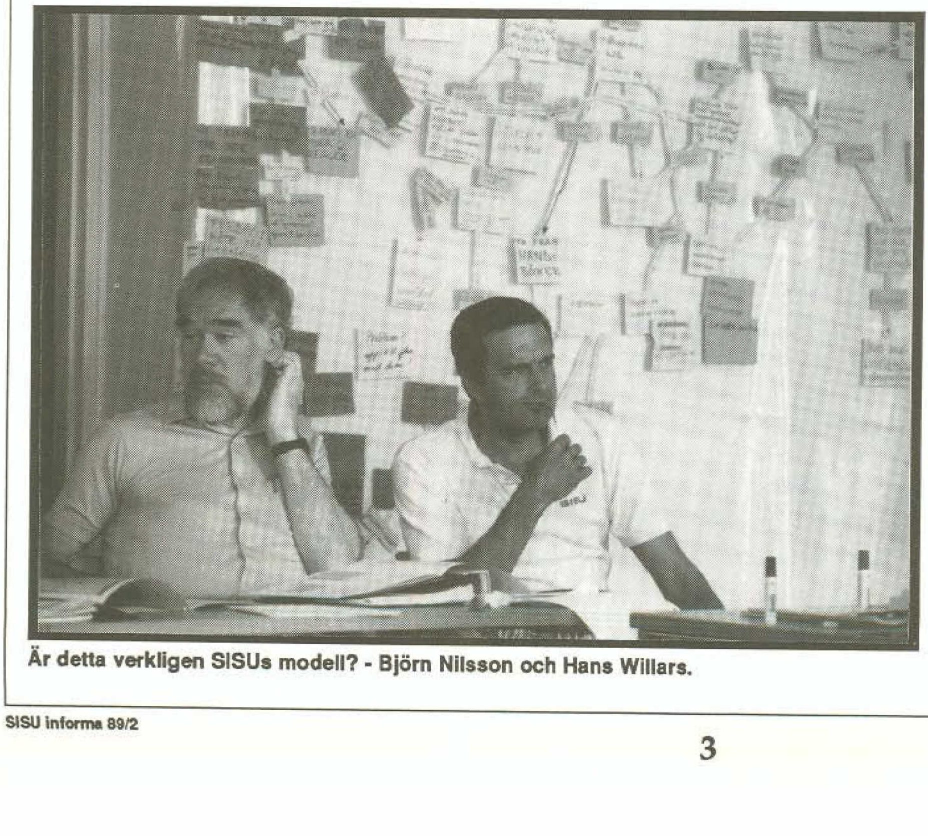
SISU modellerar SISU



Området KBS presenteras av Sten-Erik Öhlund (t.h. i bild) för fr. v. Björn Nilsson, Rolf Wohed, Jan Ljungberg, Tapani Kinnula, Lars-Åke Johansson, Erik Knudsen, Benkt Wangler som sitter, Peter Rosengren, Ulf Persson, Stig Johansson och Marianne Janning.

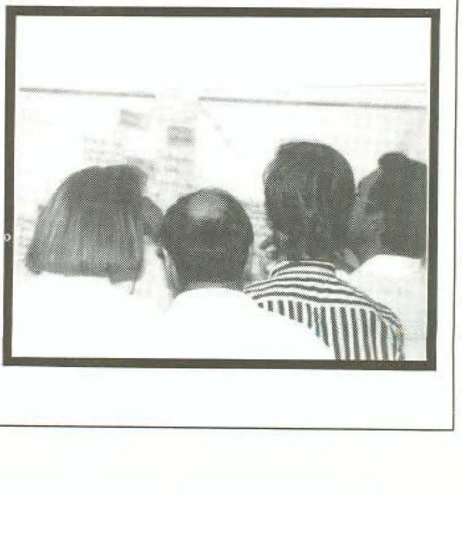
Nedan: Mats R. Gustafsson, Jonas Olsson, Ulf Persson, Roland Dahl och Lars- Åke Johansson modellerar området CASE.

Tapani Kinnula, Jan Ljungberg, Ulf Persson och Sten-Erik Öhlund på "vernissage".



Är detta verkligen SISUs modell? - Björn Nilsson och Hans Willars.

Vi vill också klistra! - Ann-Charlotte Eriksson, Benkt Wangler, Stig Johansson och Hans Willars vid "Modell och metod"-områdets modell.



CASE89: Föredrag och flickkör byggde broar mellan kulturer

Uti vår hage, där växa blå bär...

Under eftermiddagskaffet dök plötsligt en flickkör upp på en av broarna över Electrums luftiga ljusgård. Svenska folkvisor klingade mellan glasytorna, till konferensgästernas tydliga förtjusning.

Sången var ett oväntat inslag. Men i övrigt löpte det mesta programenligt på CASE89, den första nordiska konferensen om datorstödd systemutveckling. Drygt 200 deltagare kunde arrangörerna SISU och SSI räkna in.

av Lottie Eriksson

Flickkören var en del av operation dagsverke, då elever från landets högstadie- och gymnasieskolor samlar in pengar till något behjärtansvärt ändamål. I år gällde det skolor i Uganda.

Det internationella perspektivet var även i övrigt närvarande på CASE89, även om det mest handlade om den industrialiserade västvärlden. Under tre dagar kunde konferensdeltagarna lyssna på 40-talet föredrag av forskare och praktiker från tolv länder, eller besöka utställningen där ett tiotal företag visade upp sina CASE-produkter.

På programmet stod såväl presentationer av forskningsrön och teoretiska modeller som redovisning av praktiska erfarenheter. Ett av syftena med konferensen var nämligen att skapa kontakter mellan forskning och näringsliv.

Franskt om konservativa verktyg

Inledningstalare var Colette Roland, professor på Sorbonne-universitetet i Paris.

-Första generationens CASE-verktyg är mycket konservativa, sade hon.

Detta gällde främst de metoder som dagens CASE-verktyg bygger på. Enligt Rolland begränsas valet av metoder till att gälla sådana som använts i minst tio år, vilket bl a medför en konstlad distinktion mellan passiva och aktiva komponenter (data respektive processer eller händelser). Hon ansåg också att dagens verktyg mer ger stöd för att registrera och dokumentera systemdesign än för designprocessen i sig, en åsikt som för övrigt delades av flera talare.

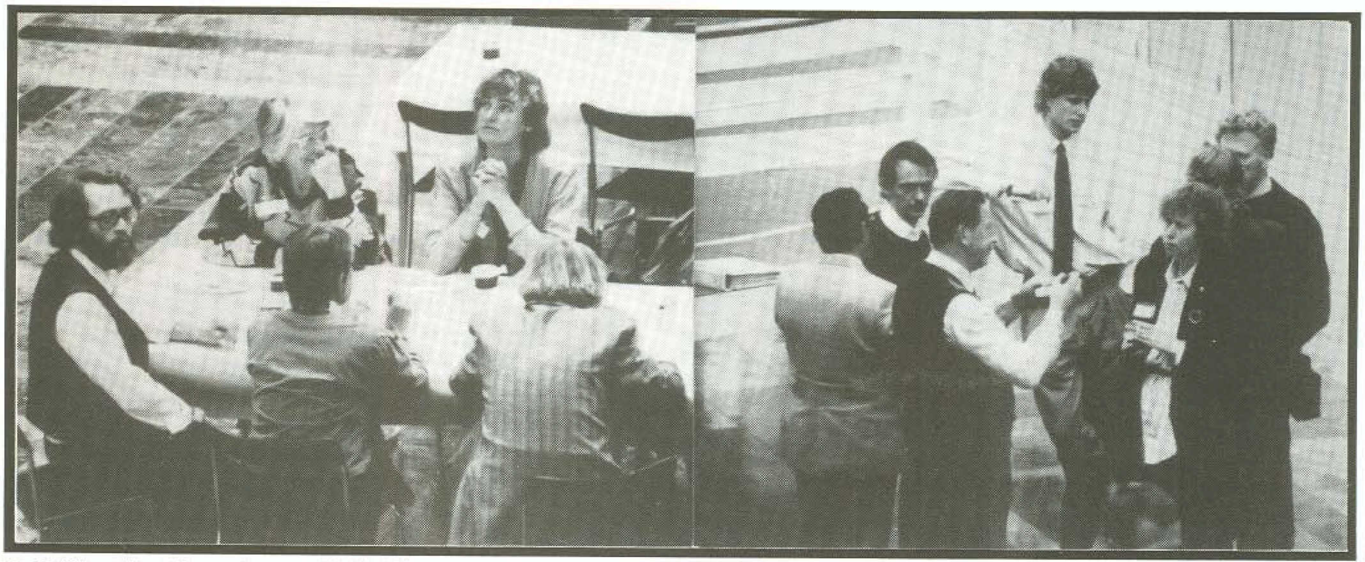
Basen för framtidens modellering, menade Rolland, måste vara objektorientering och AI-teknik, s k artificiell intelligens. Designprocessen måste formaliseras, ett område där expertsystem kan vara till hjälp. En idé som befinner sig i ett mycket tidigt skede är också utvecklingsverktyg som kan lära av tidigare utvecklade system, och återanvända kunskap vid nyutveckling.

Brittiskt om branschens brister

Flera talare betonade att databranschen måste utvecklas från att vara teknikorienterad till att ha kunskaper om den verksamhet där tekniken används. En av dem var Michael A Kingsbury, vars åsikter grundar sig på mer än trettio års erfarenhet av den brittiska databranschen. Han karakteriserade branschen som "fylld av teknologi men med ytterst litet visdom", och liknade systemutvecklarna vid ingenjörer på 1880-talet.

-För 21 år sedan myntades begreppet "software engineering", sade han. De problem som definierades då var att system levererades för sent, blev för dyra och inte tillgodosåg kundernas behov. Det är fortfarande giltigt.

Enligt Kingsbury saknar databranschen kvalitetskultur. Man koncentrerar sig på verktyg och teknik, utan att ha klart för sig vad de system man utvecklar skall användas till. En av de stora bristerna ansåg han vara att branschen tillförs för litet kunskap utifrån. De flesta nyrekryterade är unga män-



Er tillfällen till möten och samtal. Här i Electrums glasgata.

niskor som kommer direkt från universitet eller högskolor, och cheferna har ofta erfarenhet enbart från databranschen.

Kingsbury lade fram några förslag till åtgärder för att komma tillrätta med problemen, bl a att system för kvalitetskontroll skulle införas på universitet och högskolor. Han uppmanade också datavärlden att studera hur andra branscher hanterar ledarskaps- och kvalitetsfrågor, liksom att titta på hur systemutveckling bedrivs i andra kulturer än den västeuropeiska eller den nordamerikanska. Ett område som kräver forskning är likaså hur systemdesign skall kunna beskrivas på ett sådant sätt att systemutvecklare kan jämföra och dra lärdom av vad som gjorts tidigare.

Belgiskt om framtiden

Även Franz van Assche från James Martin Associates i Belgien anknöt till kvalitetsaspekter.

-Med hjälp av verktyg som 4GL, kodgeneratorer etc utvecklar vi fortfarande opålitliga system - men mycket snabbare, sade han.

En framtida effekt av CASE-teknologin kan vara att färdiga system lättare kan anpassas till kundens specifika behov.

-Idag står valet mellan systempaket eller ett skräddarsytt system, sade Franz van Assche. Med CASE kan distinktionen komma att blekna. Leverantören kan ha t ex ett ordersystem med ett inbyggt CASE-verktyg, så systemet enkelt kan anpassas.

Han förutspådde också en stark utveckling av arbetsstationer, med grafiska inslag som fönster, symboler etc.

Objektorienterade databaser nämndes av flera talare som ett givet inslag i framtidens systemutveckling.

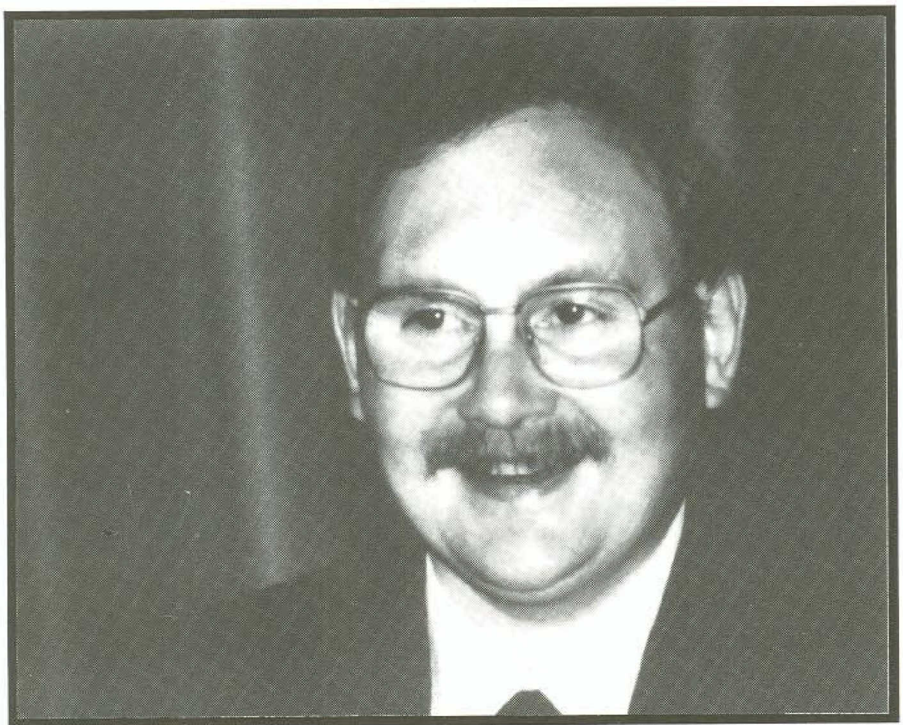
-Objektorienterade databaser finns på marknaden, men är ännu inte tillräckligt effektiva, sade van Assche. De befinner sig i ungefär samma skede som relationsdatabaser gjorde för 10-15 år sedan.

Text och bild i skön förening

Hur datorn kommunicerar med de människor som använder den blir allt viktigare. Där kan datafolket dra nytta av den kunskap som finns samlad i den grafiska branschen, som har lång erfarenhet av att presentera information så att den blir lätt att tolka och förstå. Anita Kollerbaur och Thomas Larnhed från Stockholms universitet talade om hur lexivisuella gränssnitt kan förbättra kommunikationen mellan människa och maskin.

Lexivisuell presentation bygger på ett nära samspel mellan bild och text. Tekniken är idag vanlig i tidningar, uppslagsverk, skolböcker osv. Men den används också i många andra sammanhang. Ett exempel som Kollerbaur och Larnhed visade var den för alla svenskar välkända broschyren "Dags att deklarera". På senare år har den innehållit en lexivision som visar hur sammanställningen på deklarationsblankettens första sida skall fyllas i. Med hjälp av grafiska symboler som pilar, rutor etc kan läsaren snabbt se från vilken av de övriga tre sidorna en viss uppgift skall hämtas, eller få en förklaring på vad som skall stå i en viss ruta.

Att använda lexivisioner i datasystem medför naturligtvis speciella problem, inte minst beroende på den begränsade ytan. Men med hjälpmedel som fönster, symboler och bläddringsfunktion går det att åstadkomma skärmbilder som både ger överblick, sätter in detaljerad information i sitt sammanhang och hjälper användaren att hålla reda på var i systemet han/hon befinner sig.



Simon Holloway fick toppnoteringen när det gäller deltagarnas bedömning av värdet av respektive föredrag. (foto: Petter Nylander)

För att få bästa resultat bör naturligtvis de lexivisuella principerna beaktas redan under systemutvecklingen. Men det går också att förbättra befintliga applikationer utan alltför stora ingrepp. Det visade Anita Kollerbaur och Thomas Larnhed med exempel från ordbehandlingsprogrammet MS Word, där deras forskargrupp gjort alternativa förslag till layout av några skärmbilder.

Teknisk skribent nödvändig

Det finns fler faktorer som är viktiga för hur människor uppfattar ett datasystem. Något som systemutvecklarna ofta saknar stöd för är utformningen av dokumentation för slutanvändare. ABB Data har satsat på att förbättra just dokumentationen, i ett projekt som Lars Hemingstam på Kreativ Systemutveckling berättade om. Ett av inslagen är professionella skribenter.

-Att ta fram dokumentation är ett heltidsjobb för en specialist, sade Lars Hemingstam. På ABB Data finns idag 6-8 personer som arbetar specifikt med dokumentation.

Andra inslag är standardiserad layout och struktur på dokumentation, liksom datorstöd som gör det möjligt att automatiskt plocka in skärmbilder och rapporter från applikationen eller att hämta fältbeskrivningar från en katalog.

Dokumentationsarbetet löper parallellt med systemutvecklingen. Den tekniske skribenten arbetar tillsammans med en referensgrupp, som godkänner eller föreslår ändringar i texterna. I referensgruppen ingår såväl slutanvändare som systemutvecklare.

Svensk miljonsatsning

Praktiska erfarenheter redovisade också Göran Lustig och Sven Ersson. De berättade om Spadabs satsning inom CASE-området, som beräknas kosta minst 25 miljoner kronor.

-Vi räknar med en trettioprocentig produktivitetshöjning, sade Göran Lustig. Och det är en mycket försiktig uppskattning.

Spadab inledde CASE-satsningen för ungefär ett år sedan, och har identifierat några faktorer som krävs för en lyckad CASE-användning. Att inte ändra företagets långsiktiga strategi för att passa det verktyg som valts är en sådan faktor. En annan, som kan tyckas motsäga den första, är att se över och om nödvändigt anpassa systemutvecklingsprocessen. För framtida utveckling behövs dessutom en gemensam databas där verktyg från olika leverantörer kan samexistera.

Sist, men långtifrån minst, krävs utbildning för både systemutvecklare och användare. Utbildning svarar för den största delen av projektets totalkostnad.

-Ungefär hälften av vår personal är programmerare, sade Göran Lustig. De skall växa från programmerare till systemanalytiker, och det tar år.

Liknande tankegångar framförde Simon Holloway från konsultfirman DCE i London.

-Vid sekelskiftet tror jag inte att det finns många programmerare kvar, sade han. Det vi vill ha från nyutbildade är inte längre logik och förmågan att skriva program i ett visst språk. Vi behöver folk som kan analysera och lösa affärs- och verksamhetsproblem.

Idérikt men obestämt från IBM

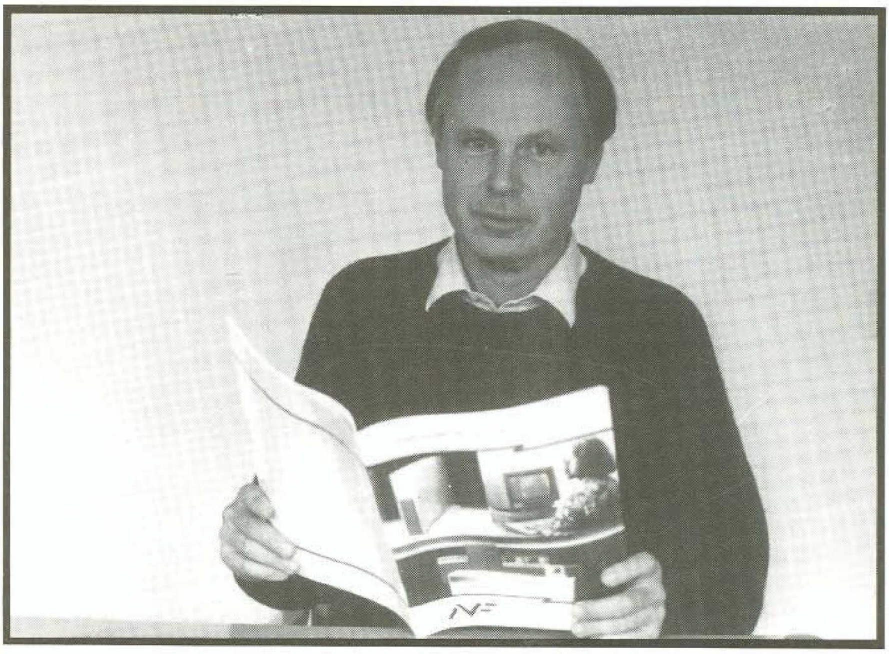
Miles Welter hade kommit från USA för att berätta om IBM:s tankar runt systemutveckling. Han sade att IBM:s ambition är att knyta samma metoder och verktyg inom en gemensam ram. Grunden för detta skall vara en databas ("repository") som gör det möjligt för verktyg från flera leverantörer att samverka. För systemutvecklaren skall gränssnittet vara utformat på samma sätt oavsett vilket verktyg han arbetar med.

Något klart besked om när detta kan bli verklighet fick dock inte åhörarna. Welters skisserade en kedja av produkter, och sade att den totala funktionaliteten kunde bli tillgänglig i begränsad omfattning inom två år efter det att den första produkten lanserats.

CASE tools in an IRM environment

De två senaste åren har Vägverket satsat mer än 400 miljoner kronor på att bygga upp en distribuerad datastruktur. I sitt föredrag på CASE89 redogjorde Jaro Potucek för Vägverkets syn på informationshantering och planer på CASE-användning. Vi publicerar här föredraget i sin helhet.

av Jaro Potucek, Vägverket



Vägverket (Swedish National Road Administration, SNRA) is a decentralized organization with a number of regional units sharing many similarities. EDP support shall enable an efficient information supply throughout the entire organization through widespread usage of common data. Data includes both data structurable in tables as well as different documents. Data is considered to be an integrated corporate resource requiring administration. IRM (Information Resource Management) in SNRA's decentralized organization is acheived by a corporate data policy, issued by the Director General, and completed by a set of steering instruments. The steering instruments are:

* common structures (for data, business functions, hardware, data storage, applications)
* standard methods (for modelling, security, system development)
* standards for hardware and general software
* information administration functions on different levels.

One of the standards for general software is the so-called System Development System (SDS), an integrated EDP tool for system developers. SDS supports also the initial stages of system development and the methods used in them. This means that SDS must include CASE tools. CASE tools in an IRM environment means that CASE tools are not longer exclusive tools for professional system developers. CASE tools are used for description of all types of structures related to informa-

The Swedish National Road Administration (SNRA)

Approximately 13,000 people work for the SNRA; of these, about 8,500 are employees. The 1988 budget was 8 milliard SEK. The SNRA administers approximately 100,000 kilometers of national roads and supervises the distribution of state grants to both major municipal as well as private roads.

Head Office with its approximate 650 employees is situated in Borlänge. In addition to the top management, six service departments with most of the specialists and research and development personnel are located there. The regional organization consists of a road administration in every county with an average of 60 employees. On the local level there are approximately 250 maintenance and operation areas with about 15 employees working at each. On the construction side, the SNRA is organized into six construction districts each with approximately 30 regionally employed people. A hundred or so construction work sites consisting of a varying number of employees come under the supervision of these construction districts.

The EDP department with a little more than 100 employees is one of the six departments situated in Borlänge. It prepares policies for EDP support on top management request as well as provides EDP support to the entire organization. It also supplies computer utilities for the Head Office as well as personnel for EDP development. These two latter services are debited internally tion handling. It means data structures, business functions structures, EDP equipment structure and program system structure (data storages and applications).

The structure descriptions are needed by user groups which are not EDP professionals. Information administrators use them for description of information effects of business planning, system administrators use them for different tasks in system administration (maintenance, planning, user support), and last but not least, end users use them for information retrieval.

Integration of information handling into business operations means that in the future we can also expect direct usage of these structure descriptions in all kinds of business development. We have identified three essential demands regarding integration:

- * integration between different types of structures
- * integration between CASE and other development tools
- * integration between CASE and applications in operation.

We have also identified three levels of functionality of CASE tools:

- * documentation
- * documentation + retrieval
- * documentation + retrieval + different controls.

Today, we meet the common conflict between vast expectations on CASE tools and their modest functionality. Nevertheless, we will start now with simple CASE tools, but use themin a flexible way. The further development will take us closer to our vision. During the time, we expect to be forced to move our structures between different tools several times.

SNRA Policy for Information Management

EDP within the SNRA is administered through a general overall policy for the entire organization. This policy states the strategic goals for EDP support, the steering prerequisites as well as the strategy to be used.

The strategic goals can be expressed as an overall efficient management of information for the various operations; that is, primary, supportive, and development. One basic prerequisite for this is an efficient information supply; i e, correct, unambiguous data and documents available for operational needs at a reasonable cost. An efficient information supply is promoted by uniformity of concepts, data and documents. Decisions on uniformity must be related to the data structure and be based on a survey of the various operations and their information needs.

The most important prerequisites governing EDP support consist of the positive opinion of the human element, the decentralized organization, and the areas of concentration indicated by top management. The positive opinion of the human element limits EDP as an aid in information management and leads to the following restrictions:

- * EDP support shall not cause any knowledge impoverishment
- * EDP support shall not lead to increased isolation, work monotony or infringe upon individual integrity
- * ergonomic factors shall also be taken into consideration (i e, health risks of different kind shall be avoided).

The strategy firmly states that information be treated and administered as a strategic resource to ensure efficient information supply. Every manager, responsible for business operations, is also responsible for EDP support to them. All people, involved in a business operation, are concerned with its decision support including EDP.

EDP within the SNRA

Until recently, EDP support within the SNRA was totally based on Unisys computers used in different types of applications. The most important of these were finance administration, personnel administration, information on roads and traffic, as well as EDP support for road design. At the regional level, there were approximately 50 terminals and 250 personal CP/M computers used both as terminals and as independent computers for less complicated applications. EDP support was, in practice, totally absent at the local level. On the whole, it could be said that EDP was used extensively; i e, as isolated systems supporting certain operations in the central administration, and, to a lesser degree, at the regional level.

The first stage of the new generation SNRA computerization program was accomplished in 1987-1988 at a cost of more than 400 million SEK. This stage provided standardized and integrated computer utilities. VAX computers were installed at the Head Office as well as at every Regional Road Administration. More than 2,000 terminals and personal computers were connected to these at the central, regional and local level. The total equipment cost was approximately 200 million SEK. Office automation programs were installed as well as 8 different applications covering the basic needs of the most important user categories at the SNRA. The applications were developed within the SNRA at a total cost of over 100 million SEK. 3,000 users have been trained at a cost of 100 million SEK.

CASE tools in an IRM environment

In the SNRA's decentralized organization the following steering instruments are used:

- **common principles for information management (handling of data and documents)**
- **structures for information management and its components**
- **standards for frequent components (e g, EDP hardware and general software) and work methods**
- **organizational safeguards in information administration (IA functions).**

Additionally, data storage and applications supplied on request of top management and used throughout the entire organization promote integration and serve as steering instruments.

Furthermore, an efficient information supply requires competent personnel with a positive attitude. This can be achieved through the following:

- users regard EDP as a natural tool for their handling of information
- users are given the opportunity to participate in the development of EDP support in order that their demands can be taken into consideration
- all personnel receive the information and training necessary
- personnel whose work tasks decrease or disappear completely are provided with the training required to be able to assume other tasks within the organization. (This applies, for example, to personnel involved in design drawing, data registration or typing.)

Principles for Data and Document Management

The backbone of information management consists of how data and documents are handled; that is, the methods used to ensure the production, storage, distribution and access to them. In effect, it requires that every manager devote special attention to the administration of data and documents within his area of supervision in accordance with the SNRA policy on information management as well as other steering instruments.

One necessary prerequisite for data to be able to be used throughout the organization is that operations, information needs, data and other important phenomena are described in a way that is structured, coherent and comparable. A consistent terminology is necessary in order to achieve this.

Concepts and data are registered in a definitions dictionary available throughout the entire organization. References shall provide information as to who is responsible for specific data and where it can be obtained or ordered. Documents concerning the entire organization are to be listed in a reference data base available to all SNRA employees. The references shall provide information as to who is responsible for a specific document and where it can be obtained or ordered.

Uniformity with Respect to Data and Documents

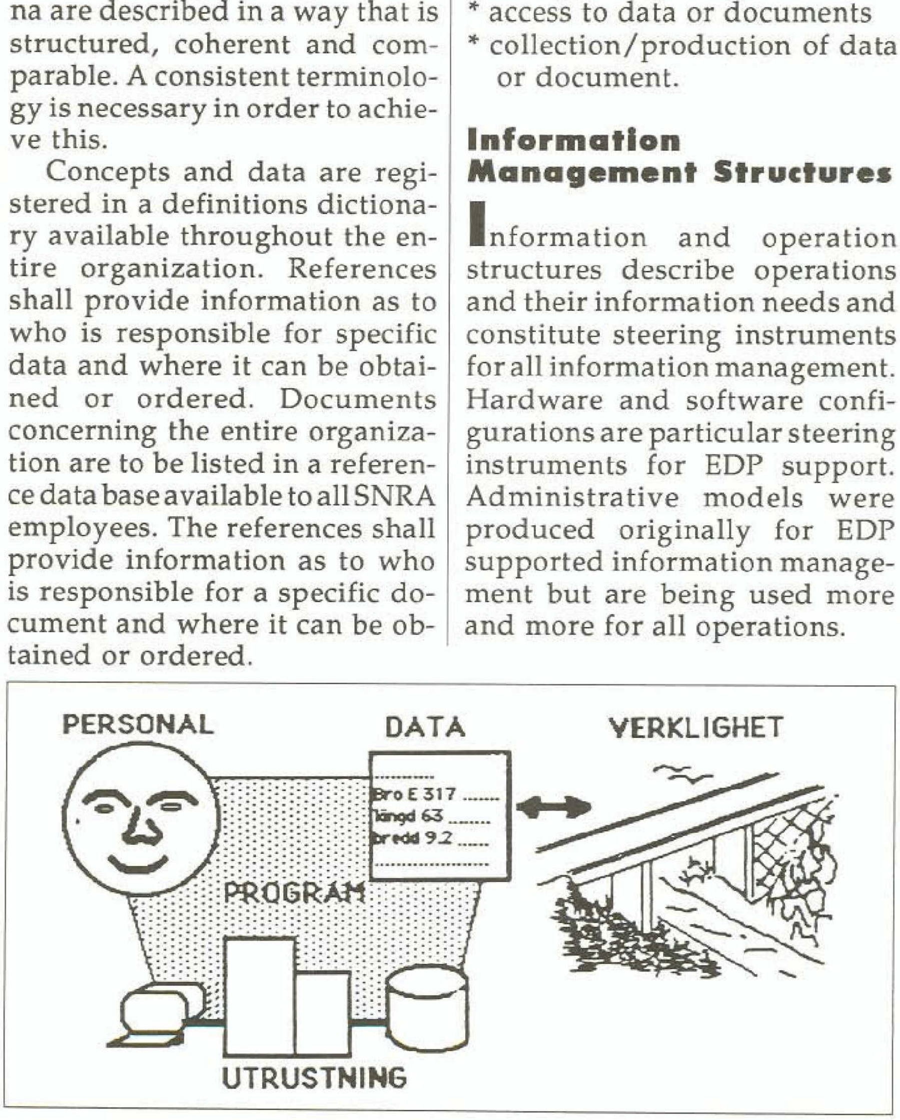
Concepts, data and documents shared in various operations must have an unambiguous common interpretation. This is acheived through a decision on uniformity establishing their status and description. Decisions concerning uniformity must be related to descriptions of the operations. Uniformity decisions shall be made by every individual manager for his/her particular area of operation.

Uniformity of concept, data or document can include one or several of the following:

- definitions of concepts, data or documents
- storage of data or documents
- access to data or documents
- collection/production of data or document.

Information Management Structures

Information and operation structures describe operations and their information needs and constitute steering instruments for all information management. Hardware and software configurations are particular steering instruments for EDP support. Administrative models were produced originally for EDP supported information management but are being used more and more for all operations.



CASE tools in an IRM environment

Operation Descriptions (Operation Structures and Information Structures)

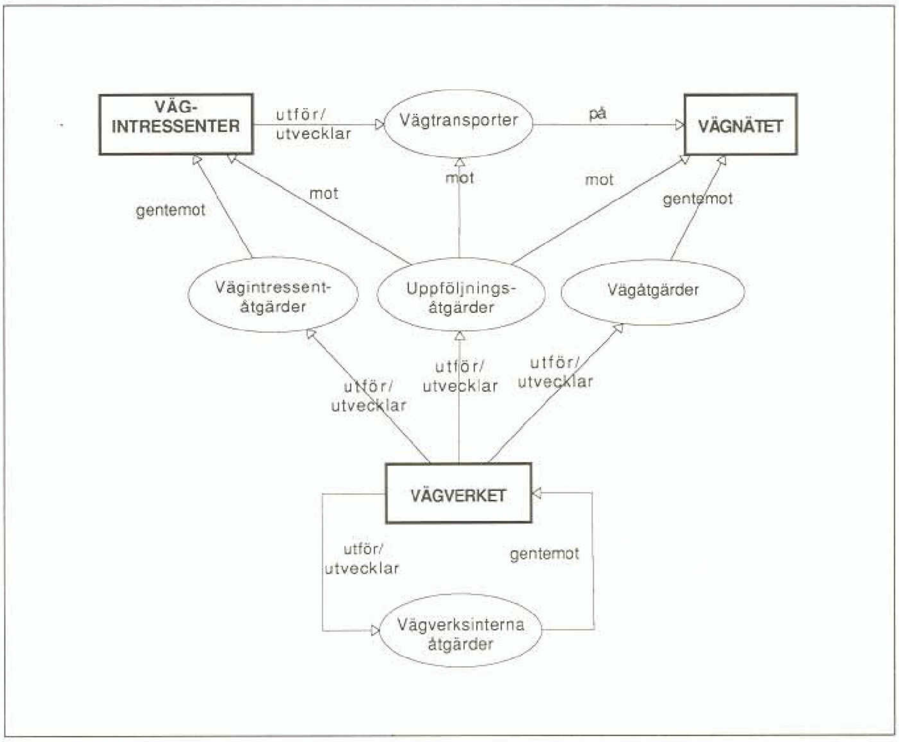
There are to be current descriptions of all operations at the SNRA. These operation descriptions shall represent the collective views and accumulated knowledge of all personnel involved. Operation descriptions shall be freely accessible to everyone concerned.

Operation descriptions from different units shall be available for comparison and compilation into an overall picture. The ABC method shall be used as a common descriptive method.

The operation descriptions shall be produced gradually, normally as a component in the work on systems development or organizational development.

It is always the manager in charge of a certain area of operations who is responsible for the factual content, quality and current relevancy in the operation description.

Operation descriptions shall always constitute the basis for both the development and administration of EDP. This is especially important in the design of the application structures and data storage as well as when making decisions on the overall policy on concepts and data. The operation descriptions shall also be a natural tool in operation management, organizational development, internal and external information and in other contexts where data and documentation questions are involved.



Computer Utility Structures and Standards for EDP Tools

In functional terms, the SNRA has a uniform computer utility structure. There should at all times be a number of specified standards for EDP hardware and general software (so-called EDP tools) which together cover the functional requirements of the computer utilities structure in the best possible way. Both the purchase and development of EDP hardware and software follow these standards.

Program Systems Structure (Structure for Data Storage and Applications)

There is to be an established structure for SNRA data storage and applications to which every new or altered data storage or application shall be connected. A differentiation can be made between general systems for everybody (termed office automation systems at the SNRA), common data bases, and applications for specific users.
Structures for data storage are based on a division of the total data needs in the various subject areas while taking into consideration both the geographical aspects throughout the organization as well as the applications currently in use. The subject areas are contained in the information structure. The design and localization of the data storage shall ensure both the possibility of an overall view and data integrity, as well as storage and access efficiency. The long-term data needs within the organization determine the division of data into logical data storage. The extent of and physical localization of these is governed by the actual applications.

The applications structures are based on identifiable groups or situations requiring EDP support contained in the organizational structure. General program modules are developed for frequent data process routines to be used in all relevant applications.

CASE tools in an IRM environment

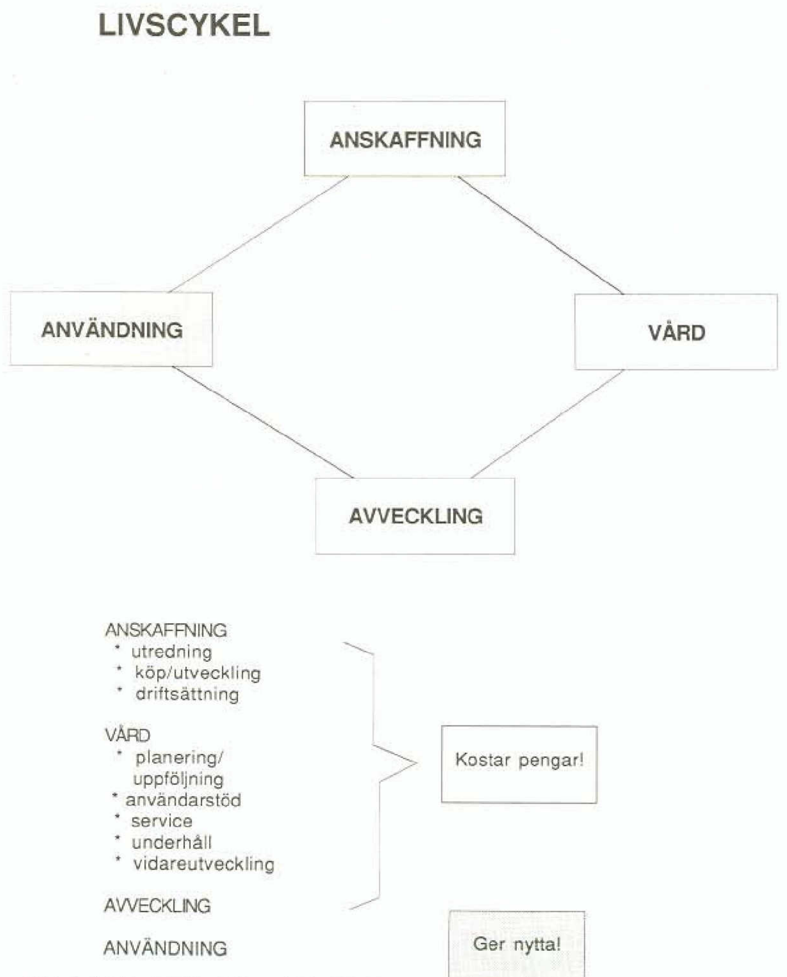
System Administration Model and System Development Model

At the SNRA we have defined system administration as maintenance, user support (including training, information and advicing), follow-up and initialization of further development of a system. System administration model incorporates a specific system development model. Related to these models is a description of the roles and responsibilities in the EDP operations. All EDP operations at the SNRA must abide by these models and role and responsibility descriptions. By the way, the models are used also for other types of resources than information systems.

Standard Methods and Standards for EDP Tools

At the SNRA there are standardized methods for both operation descriptions (the ABC method) as well as for work concerning EDP security. These will be supplemented with standardized methods for system development and other important elements of systems administration.

Up until the present time the ABC method has been used primarily in system development projects resulting in a couple of hundred concept and flow models. These models are currently documented in simple Macintosh graphs (MacDraw). The definitions will be included in a text data base and thereby retrievable.



Livscykelmodell nivå 1 Bild 1 LIVSCYKEL

Within the EDP tools area there are standards for:

- computers, including basic software
- computer operations system (aids and routines for computer operations)
- data communications
- work stations (including terminals, personal computers, printers, plotters)
- system development system (aids and routines for system development)
- office automation system (software for general office routines).

These standards have been used in Stage 1 of the SNRA computerization program but are now being supplemented to cover future needs. The most important supplements are concerned with EDP support for computer operations, ergonomics, work stations with computing power, graphic interface, more efficient development tools and CASE tools.

System Development System

Both managers and users must be given the opportunity to influence the EDP systems design both during the developmental as well as during the administrative stage. This can basically be accomplished in three ways: as clients, as user advisors in development projects or through direct involvement in the development. User influcence will depend on the type of system, and on the extent of the development effort.

Development of extensive complex systems will, in the future, continue to be carried out principally by specialists. In these cases the system development models and methods must guarantee user influence. The prime task of an efficient development system is to keep costs down.

The development of less extensive systems or of supplements to larger systems is carried out advantageously by persons whose experience has been gained at the SNRA business operations as well as in the field of system development. From an EDP point of view, such people are called "semi-professionals" at the SNRA. They have acquired their business competence through their primary job, and their system development competence through 8-10 weeks of formal training. This latter competence is maintained and further developed through using such personnel 30-50% of their working time to system development within their own specific area of operations. One major added advantage of this is the not having to wait for assistance from the EDP departement.

Moreover, end users can perform simple developmental contributions for themselves; they can design and retain calculations, design reports using report generators or diagram graphics, or even create their own applications with the aid of application generators. The advantages here are both the ability to adapt the system to specific individual needs as well as not having to wait for assistance for trivial developmental contributions.

In order to achieve this ideal situation we need both open systems as well as development system with tools for both professionals, semi-professionals and end users. The development tools must be integrated in order to get integrated applications. Training possibilities and support functions must be linked to the development system. According to those needs we have specified the so-called **System Development System (SDS), an integrated EDP tool for system developers**. There are three partly different versions of SDS for different target groups:

* **SDS-pro** for full-time professional system developers (personnel working mostly in the EDP department. These can also be external consultants.)

* **SDS- semi-pro** for semi-professional developers (personnel working with system development and maintenance about 25-50% of their time)

* **SDS for end users** reflecting the usage of OA-tools (Office Automation) for system development.

Within the SNRA we work in an integrated environment with VMS mini computers from Digital and MS-DOS personal computers from Nokia. The development system must therefore incorporate tools for both these environments. Both environments can co-operate in a single application: the data base and processing parts can be designed for the mini computer while the user dialogue is designed for the personal computer. The development tool must then facilitate a flexible data exchange. Developmental efficiency is favoured if integrated development tools exist in both the mini computer and the personal computer.

The SDS is based on **several integrated tools** for the implementation of major system development functions working under the operative system VMS:

* data dictionary (CDD)
* relational data base handler (RDB)
* screen handler (TDMS)
* report generator (Datatrieve)
* graphical kernel system (GKS)
* programming languages (COBOL, FORTRAN, JSP-COBOL pre-compiler)
* end user tool Teamdata (personal data base, report generator, query language).

As a complement to these tools we have developed a menu handler, an output handler and some general program modules.

The SDS was further completed by several, to some extent integrated, specialized tools:

* personal data base / spreadsheet / query language (Teaminfo)
* advanced statistical functions and advanced business graphics (SAS)
* text data base handler (TRIP, TDBS/A1).

We have also specified some other functions for which we need to purchase or develop development tools:

* application generator (4GL)
* graphical user interface (Windows)
* three-dimensional design (CAD)
* geographical information system (GIS)
* road data base handler (our own development).

SDS is supposed to support the initial stages of system development and the methods used in them. This means that SDS must

Action Plan for the Acquisition of CASE Tools

The present course of action (the dates are preliminary)

1. Selection. Ready July 1, 1989

- * Production of requirement specifications for CASE tools and harmonization of these with conceivable methods and requirement specifications on EDP tools.
- * Theoretical harmonization with the market supply: Ramatic, Excelarator, Programmers workbench, IEW, Deft, Speedbuilder and any possible others.
- * Rough selection.
- * Practical tests on SNRA material.

2. Purchase and adaptation

(can be done gradually).
First version ready November 1, 1989.

3. First version operational.

December 31, 1989.

include a CASE tool. Its most essential functions are an extension of both a data dictionary and a modelling tool. The CASE tool is supposed to be used for system specification irrespective of the tool used for system implementation. It means that we must choose a flexible CASE tool, perhaps quite a simple one. It also means that we cannot expect full integration between CASE and other development tools.

For MS-DOS personal computers we have at present the following set of development tools:

- * relational data base dBase III and Foxbase (compiler for dBase)
- * personal data base Rapidfile for end users
- * traditional programming languages COBOL, FORTRAN
- * Spreadsheet 20/20 and Supercalc 4
- * diagram graphics Graph-in-the-Box
- * menu handler PC-menu
- * terminal emulator and file transfer to VAX Reflection 2+.

Information Administration and its EDP Support

The various structures create prerequisites for an integrated information management which, in turn, makes common data usage possible. Naturally, the structures must first be produced and then administered. The administration comprises maintenance and further development, training, information and advisory service, as well as a follow-up of use and effect. This work is termed Information Administration (IA).

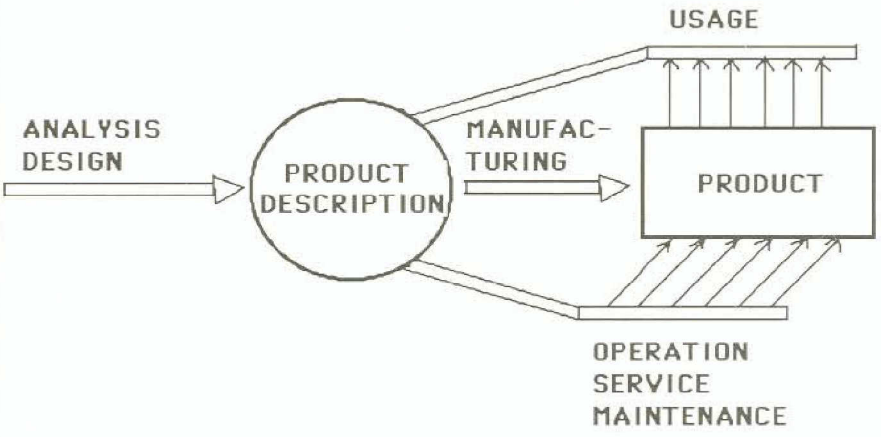
Information administration is carried out on different levels. General overall structures hold the different operations together and are administered by a corporate IA policy function on behalf of the top management. Nevertheless, every manager is responsible for the production and administration of structures within his own area of operation. These are based on higher level structures but are more detailed.

Information administration can be regarded as a first stage in the system development process. Before an application can begin to be developed, it must be "placed on the map"; i e, set in relation to the various structures. Data to be processed is specified in the information structure. The business functions to be supported by EDP are indicated in the organizational structure. The computer utility structure specifies the hardware to be utilized and the program system structure indicates the relation between the different data storage and the applications. Finally, the administration model specifies how the applications shall be developed and administered.

Information administration within the SNRA as a whole is complicated and thereby inconceivable without EDP support. EDP support is also mandatory for the co-ordination of information administration at the different organizational levels. The role of information administration as the initiator of system development places additional requirements on a close connection between development methods and the development system.

EDP support for information administration is usually called CASE tool, (CASE = Computer Aided System Engineering) or Data Dictionary (in a very broad sense of the word). CASE tools can, however, also be used by other user categories and in other connections than in information administration.

CASE tools in an IRM environment



ANALYSIS+DESIGN result in PRODUCT DESCRIPTION.
PRODUCT DESCRIPTION is used during:
- MANUFACTURING
- OPERATION+SERVICE+MAINTENANCE
- USAGE.

CASE Tool Users

CASE tools are used to handle the corporation data encyclopaedia (data on data, metadata). They will thus be utilized by various groups of users with highly varied work tasks and experience in EDP support. The most important user groups are:

* Information administrators at the Head Office and in the line of command constitute the main users of CASE tools. These are relatively few in number, are very familiar with EDP support and place higher demands on the tool functions than on their being user friendly.
* System designers with high EDP competence, who also place higher demands on functionability than on user friendliness.
* System administrators whose primary competence lies in their experience in various business operations but who are also relatively familiar with EDP support and its uses.
* Users of EDP support who require tools that are easy to use and adaptable to their operation needs. Managers can also be included in this group.

Functional Requirements on CASE Tools

CASE tools shall facilitate the description and handling of information management structures; i e, their creation, storage, distribution, recovery, presentation, consistence control and comparison of different models. The most important requirements are concerned with simple creation, simple modification, fast recovery and clear presentation.

The description of the information structure shall amongst other things comprise:

* the relation between concepts (in form of structural images)
* the relation between structural images with different degrees of detail
* the connection between concept definitions and structural images
* the connection between attributes and concepts.

The description of the business operation structure shall amongst other things comprise:

* the relation between different business functions in the form of different flows - products, material, information, money (in form of structural images)
* the relation between structural images with different degrees of detail
* the connection between function descriptions and structural images
* the relation between concepts and operational functions.

The description of the computer utility structure shall amongst other things comprise:

* the relation beween the different hardware (in form of structural images)
* the connection between the description of the hardware and the structural images
* the connection beween the hardware and the business functions.

The description of the program system structure shall amongst other things comprise:

* the identification of the different data storages (connection to the information structure)
* the connection between the data storages and the computer utility structure

- * the connection between the applications and the computer utility structure
- * the connection between applications and data storages
- * the connection between applications and the business functions.

There can be different levels of ambition with respect to CASE tools. The lowest level of ambition permits only a documentation of the various information management structures. The documentation simply defines and describes the components of the structure and the relation between them. The relation can be presented in the form of tables and/or graphs.

At a higher level, demands can be made on efficient retrieval functions facilitating manual control of logical relationships.

At an even higher level, a certain degree of help for the information administrator can be otained. This help can constitute the production of clearer structural images or of logical checks.

CASE tools shall also be able to be used for assistance in navigation within the structures. If the tools are user friendly, they can even be utilized by end users thereby serving as aids in the design of menus in the user dialogue. In certain cases these tools could even replace the user menus.

Our vision for the future is to have CASE tools to support all phases of information management, from business operations analysis and the planning of business development to the useage and retrieval of data in the final information processing system. The tools would also be able to be used for administration, operation and maintenance. A system of information exchange implying that data registred in one context could be readily transferable to another would be a reality. A comparison of data collected by different tools in different circumstances could also be made.

Integration Requirements on CASE tools

Integration requirements apply partially to the **integration between** the different **structures**, partially to the integration between **CASE tools** and **the system development system** (including **the methods** supported by the system development system), and partially to the integration between the CASE tools and the **EDP applications**. The integration must bridge the gap between the different **hardware, operative systems, models, terminology, data storage methods, programming languages etc.** These requirements can partly be met through the CASE tools themselves. However, extensive supplementation is probably a necessity.

Integration between the structures will make a description of the relation between the different structures possible; e g, between data and data storage, data storage and applications, data storage and computer utility, application and business function etc. The most important of these types of relationships are stipulated in the functional requirements.

The integration between CASE tools and the development system / development methods will primarily make it possible to use the data structures documented in the CASE tools for the construction of data storage as well as to use the standard program modules documented in the CASE tools for program development.

The integration between CASE tools and applications will allow changes in the structures to have an immediate influence on applications (e g, active data dictionaries). This contributes to a faster and more reliable system maintenance.

CASE in the Planning of Business Development

Business development comprises development both with respect to the primary business functions as well as to support (e g, support production, administration of various resources etc). It happens more and more that the business development brings about an expansion of the need for information supply and that some form of EDP support will be necessary.

When planning business development, the following must be documented:

Business goals

These are often presented through graphs with definitions. Exactly how depends on the methods used. There is no one single prescribed method for describing goals; several are possible. In the work on goal description it can be necessary to make comparisons with results of previous work within a specific field and even with other related areas.

Steering Prerequisites

The description points out the most important external and internal factors influencing the specific business function. The description must be done explicitly in writing but may also contain certain illustrations.

Description of the Business Function

The description shall refer to both the present situation as it actually is, as well as to the situation desired. The documentation shall thus include two parts comprising:

- either a flowing text or information as to where such text can be found
- conceptual models
- flow models
- facts on quantity; e g, volumes, frequencies etc.

Conceptual models define and identify the concepts used in the business function and describe their inter-relationships. Flow models identify and define functions as well as the different flows between them. Every conceptual model and flow model is part of a situation and describes a part of the total business. Each one may be linked to a certain part of the organization, but does not necessarily have to. The models as such have a certain status; i e, they describe the business function in a given situation, for example, the present, the future, or as an idea.

Business Functions Information Needs

This is a rough, formalized description of the information needs which have been identified during the creation of models for the desired situation.

Extensive and complex amounts of information are handled when planning bussiness development. EDP support is a necessity for satisfactorily coping with this work.

Business development is planned both on a continual basis as well as through special investigations. Results must be documented similarily, preferably using the same tools in every situation.

CASE in System Development

Development of the EDP systems (applications) at the SNRA is done according to the SNRA system development model. In this model, the development has been divided into the following phases: initiation, analysis, design and construction.

Information on existing applications and the business functions they support is required all the time when developing applications. Results from different business descriptions must be compared. Furthermore, information needs must be checked against different data models in order to be able to discover any possible similarities. As a result, existing data and existing program modules can be re-used

The first phase, initiation, is closely related to the planning of business development. It is during this phase that ideas on EDP support are described - ideas forming the basis for decisions as to whether aquisition will be through purchase or development. During this stage, a rough description of the involved business functions is done in order to set limitations for the work to follow. A rough business analysis is made for the area in question. Prevision descriptions of different business functions, work routines, information requirements, basic concepts dealt with, and their connections are used.

CASE in System Administration

Administration of an application means responsibility for the continued existence of the application through maintenance and initiation of development, for the efficient use of the application through training, information and advice, and for follow-up on the functionality, use and benefit. The individual administrator must have access to extensive documentation on the application in order to be able to exercise his responsibility.

In the maintenance of an application, it has to be obvious how a change in one part of the application effects the rest of it. For example, it must be evident which programs and screen images will be affected if the size of a term is changed. Moreover, it is necessary to be able to determine whether other applications will be influenced by the change in mind.

Ideas on further development and/or termination must be analysed from the point of view of their effect on other applications, of their function in the business, on the inherent changes in the information flow, etc. The work in these situations is comparable to that done in the initiation phase and the demands on support instruments are basically the same.

The administrator must be able to advise the users on the use of the application and be able to answer questions on the information processed by the application. For that reason, he

CASE tools in an IRM environment

In the analysis phase, both business and information analyses are made. The models produced in the initiation stage are processed and examined in detail. Information requirements are supplemented with demands on access times, frequencies and quality. The information bearing concepts are defined precisely, completed with identification terms and other terms, and the relations between them are analysed. The results from the initiation phase must be available in such a form that processing and detailing can be done without having to register previous results again using new tools. In the analysis phase different suggestions and solutions shall be presented and analysed. EDP support for analysis and comparison of different models is here necessary in order to see the various alternative divisions in the subsystems. At this point it is also important to have an EDP support to check and verify the different models so that the descriptions are correct and consistent.

In the design phase the model from the analysis phase is to be defined more exactly in a normalized data model with tables, data values and rules. Work routines are specified and all information requirements checked against the data model to ensure that the information required can actually be obtained. At this stage we are approaching a traditional data dictionary and the necessity of transfering information from the analysis and documentation tools used up until now to the data dictionary connected to the data base handler in mind will be a tangible reality. Possibilities of transfering definition texts etc to any prospective help functions in the future applications will also be needed.

In the construction phase, applications described in the design phase will become the reality. This means amongst other things that the normalized data model shall be optimized in accordance with the prerequisites in the actual data base handler. It is then natural to use the data dictionary related to this, in our case CDD+. Rules and events for different concepts should be documented at this time in such a way as to be easily used when producing both the programs as well as support texts and handbooks.

must be able to find out which concepts the application handles, which business functions the application supports, the quality of the data in the data base, etc.

CASE in Information Retrieval

Three different situations are imaginable when searching for information in both traditional and text data bases:

* the person searching knows where the data is stored, he is used to searching for this data and wants it presented in the way he normally gets it from the application he always uses
* the person searching knows that there is data stored but does not know which application to use to access it and/or does not know how the data can be presented in the application
* the person does not know if there is any data stored.

In the first case, there is no necessity for retrieval support instruments. In the other two cases a retrieval support instrument is needed that is identical to, or based on, CASE tools.

In order that the person searching can judge the usefulness of the data obtained, he must have access to their definitions as well as information on their quality and relevance.

An example of a question could be, "How high have the costs for pavement work been during 1989?". In order that the person posing the question can know that the answer he receives is relevant, he must have information on several points; for example, on the definition given to "pavement work" in the data base from which the answer has been obtained, on the way costs have been calculated (are all types of costs included?), on how up-to-date the data is (what are the most recent months included?) etc. An answer that there is no data requires further analysis. Does the answer in effect mean that there is no data, or does it mean that the question was formulated in the wrong way? Information on quality, relevance and definition of data must, therefore, be documented so that it can be decided whether or not a question is worth posing.

Another type of question is concerned with definitions of concepts. An investigator or someone responsible for a business function can wish to know how "road" is defined at the SNRA and if the same definition applies throughout the entire organization. He could, for example, want to know how productivity is calculated in different applications. In these cases, the questioner is probably not interested in specific values, but rather is satisfied with the information available in the data dictionary.

CASE Tools and Document Data Bases

The foregoing has been concerned with the use of CASE tools for handling data structured in table form. However, a great amount of the data in a corporation is contained in documents; i e, flowing text sometimes including illustrations. It is naturally desirable that CASE tools also be used for information retrieval from documents.

The SNRA computer supported document handling system comprises:

* a reference data base containing information on documents of common interest. The reference data base provides references both to documents stored in computer readable form as well as to other documents. The content (in the form of short summary, keywords and subject areas) as well as the storage place is described for every document.

* full text data bases with computer stored documents of common interest

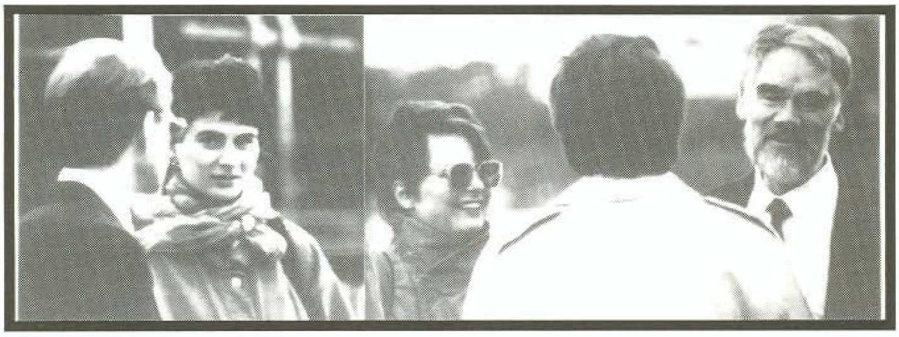
* a record data base with information on incoming and outgoing documents

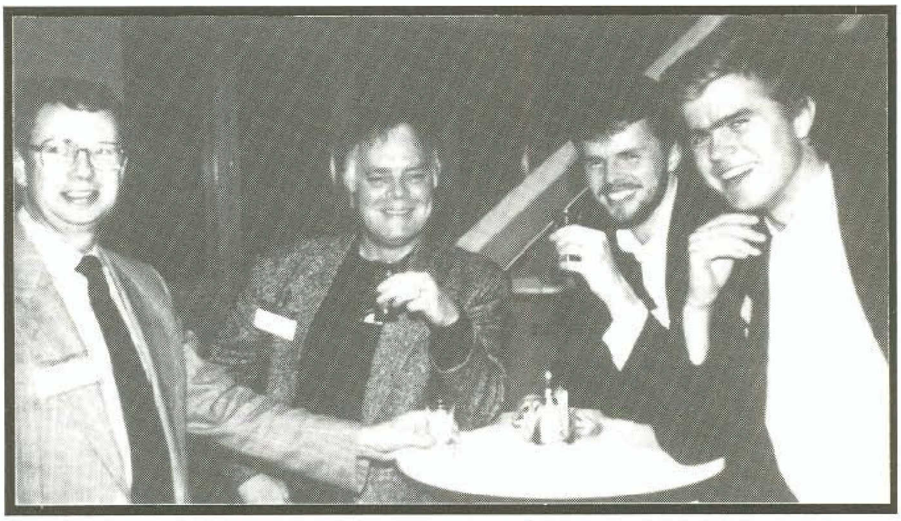
* a library data base with information on the literature inventory there.

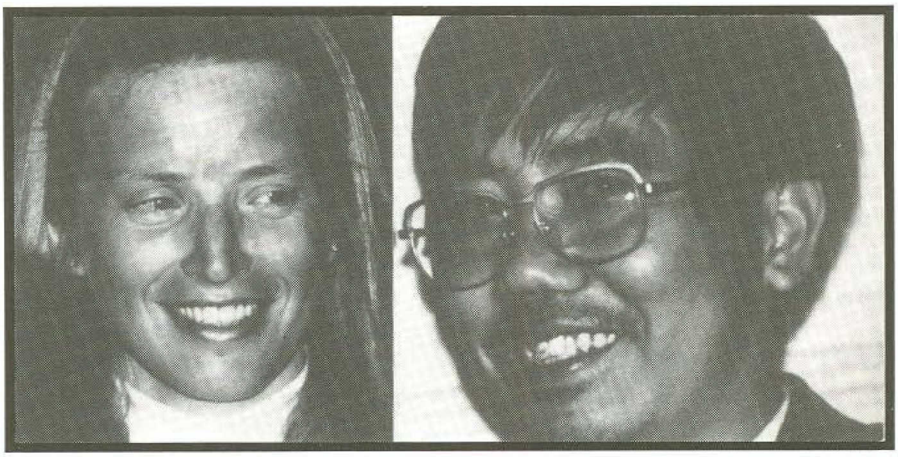
The selection of keywords used in the reference data base is currently free and unrestricted, but in the future a specific pre-determined keyword list, a so-called thesaurus, will be used. This list shall be based on concepts defined and listed in the data dictionary produced using CASE tools.

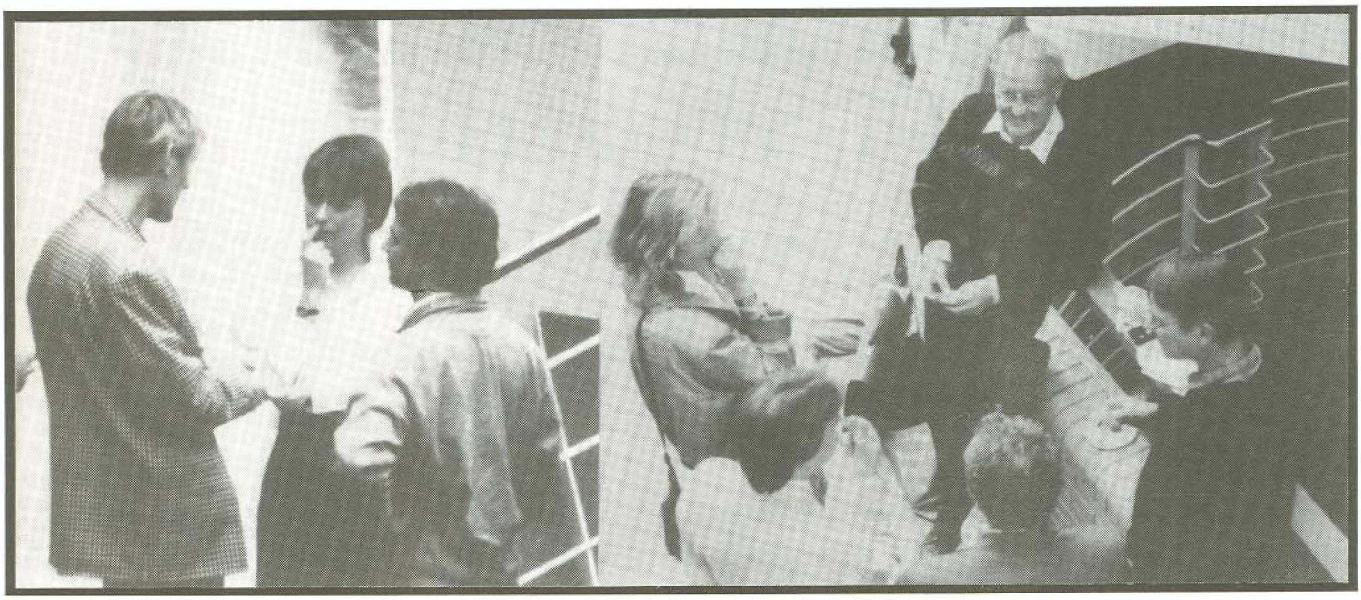
The current subject area classification in the reference data base has been done according to two older classification systems. These should be supplemented with the subject areas in the data dictonary.

When this has been done, the person seeking information will be guided by CASE tools to the correct data irrespective of whether this is stored in structured tables, computer readable documents or other types of documents. ■



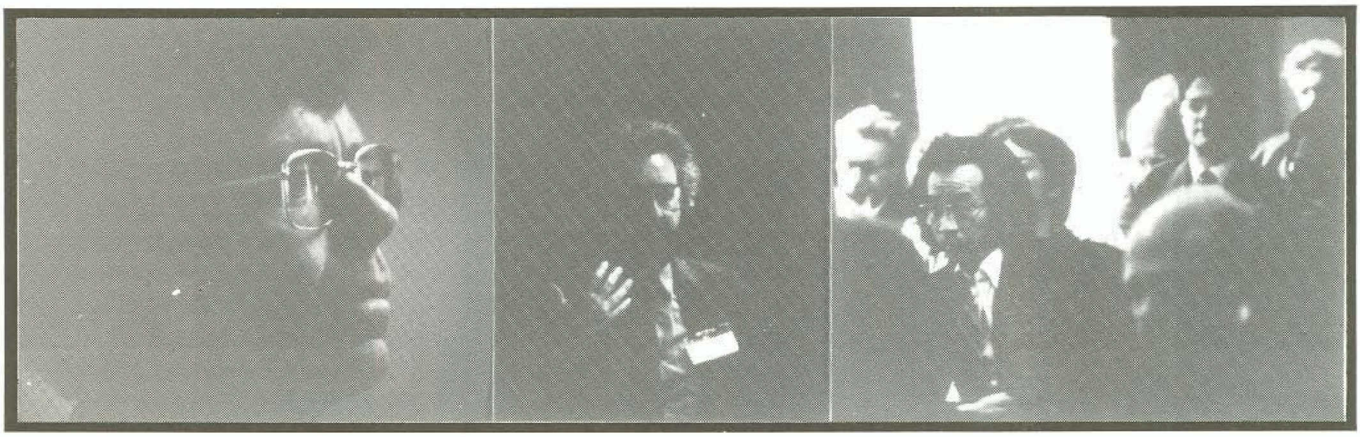


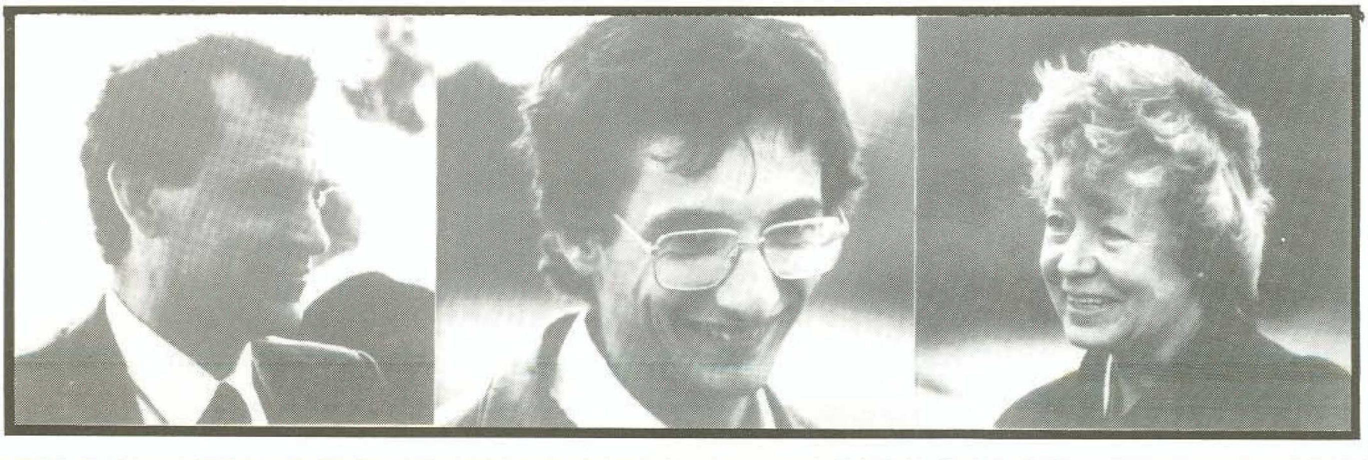




CASE89 – deltagare och föreläsare

Foton: Petter Nylander och Lars Bergman





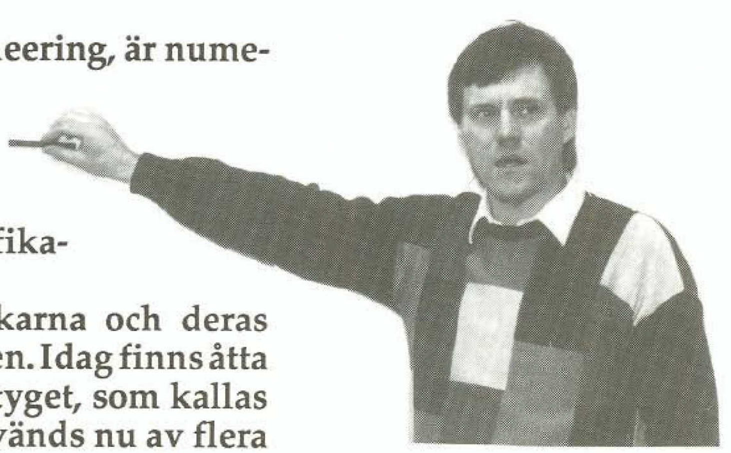
Intressenterna styr SISUs CASE-utveckling

Intressenternas behov styr SISUs CASE-utveckling

CASE, computer aided software engineering, är numera ett hett begrepp i databranschen.

Men redan innan begreppet hade myntats började ett par forskare på Chalmers arbeta med något som då kallades modelleringsstöd eller specifikationsverktyg.

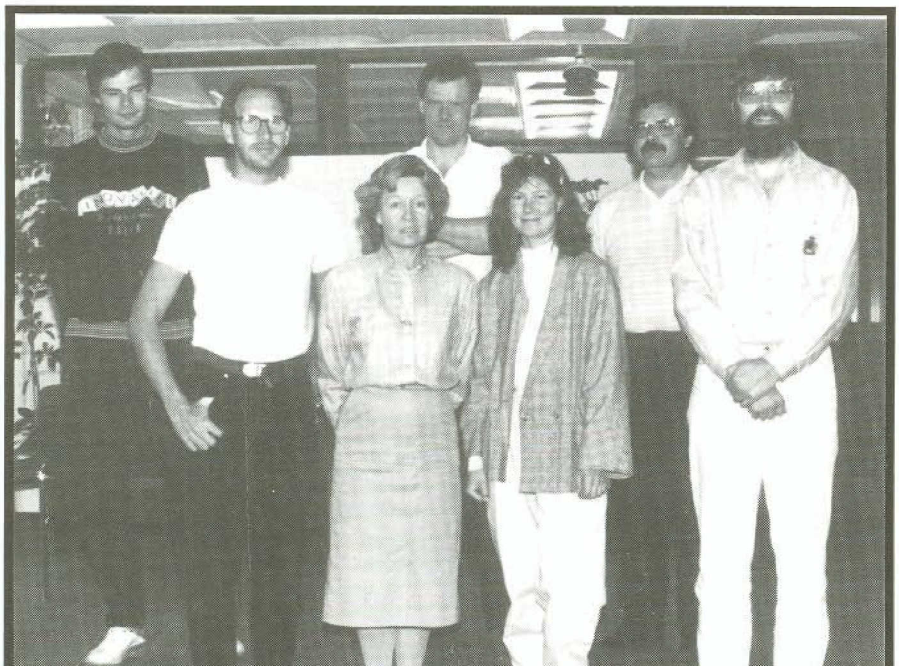
När SISU startade 1985 kom forskarna och deras arbete att bilda basen för CASE-gruppen. Idag finns åtta personer på Göteborgskontoret. Verktyget, som kallas Ramatic, har utvecklats vidare och används nu av flera av SISUs intressenter.



SISU har mycket aktivitet inom det heta CASE-området.

av Lottie Eriksson

Samling i SISU:s Göteborgskontor. Fr v: Jan Ljungberg, Jonas Olsson, Rose-Marie Johansson, Lars Åke Johansson, Harriet Dahlgren, datalingvist som ex-jobbar, Mats R. Gustafsson och Per Bergsten. Stig Johansson var på Volvo PV. (Foto: Jonas Olsson)



Lars-Åke Johansson är chef för CASE-gruppen och en av dem som varit med sedan arbetet med Ramatic startade på Chalmers.

-På marknaden finns CASE-produkter som innehåller både ett verktyg och en metod, säger han. Vi har vänt på det och bygger ett verktyg som kan anpassas efter en viss metodik. Ramatic är inte ett verktyg, utan flera.

Ett av CASE-gruppens största projekt just nu är ett s k tillämpningsprojekt tillsammans med Volvo Personvagnar, Data Logic och Digital Equipment. Projektet kallas VDDS, efter de deltagande företagens initialer.

Ökad kunskap hos alla parter

Ramatic passar väl in i filosofin bakom SISUs tillämpningsprojekt, som bygger på ömsesidig kunskapsutveckling hos deltagande parter.

De flesta företag som satsar på CASE köper ett färdigt verktyg och får utbildning av leverantören. De av SISUs intressentföretag som vill använda Ramatic får själva vara med och utforma verktyget. Det kräver mycket arbete, men ökar också metodkunskaperna.

-I projekten bygger de på sig en metodik, säger Lars-Åke Johansson. Man går igenom kvaliteten i metoden och förbättrar den. Det är behovsstyrd CASE-utveckling, verktyget formas efter läget i organisationen och efter i vilken riktning man vill gå. Organisationen får inte bara ett verktyg, utan vet också vad den behöver och vilka krav den skall ställa på CASE.

Ramatic är i dagsläget främst ett stöd för specifikationsarbetet. Det gäller t ex arbetet med att analysera, bygga upp och sammanställa specifikationer. Men också komplexa samband mellan specifikationsdelar. I den version av Ramatic som utvecklas för Volvo Personvagnar kan man t ex jämföra termer mellan verksamhetsmodeller och datamodeller eller enkelt se vilka objekt och relationer som hör till en viss funktion.

Sprida erfarenheter

När Ramatic anpassas till en ny metod definieras modelleringsbegrepp och deras grafiska symboler, menyer samt vilka formulär som används och vad de skall innehålla. Dessutom definieras en "syntax", dvs hur symbolerna för olika designobjekt får kopplas ihop.

Ramatic kan sägas bestå av en mängd byggklossar. I takt med att fler versioner utvecklas - idag stöder verktyget 6-7 metoder - ökar mängden generella byggklossar. Resultaten kan återanvändas i nya projekt, och marginalinsatserna för att modifiera verktyget till en ny modelleringsteknik minskar.

Genom SISU kan erfarenheterna från tillämpningsprojekten kanaliseras och de generella resultaten delges andra intressenter. Genom SISUs arbetssätt kan man också förmedla kontakt mellan intressentorganisationer som kan ha nytta av att utbyta kunskaper och erfarenheter. Nyligen etablerades CASELAB som ett forum för att öka och sprida kunskapen om metoder och verktyg.

Lars-Åke Johansson påpekar att det är viktigt att tänka på att CASE inte är något isolerat.

-En CASE-satsning handlar inte bara om verktyg, utan också om till exempel kunskap, metodik och arbetssätt, säger han.

Kompetens på många områden

Ramatic dominerar CASE-gruppens arbete, där man förutom VDDS-projektet samarbetar med bl a Ericsson Telecom och Televerket Stockholm. Andra inslag är dataadministration och administrativ utveckling. Inom gruppen finns kunskaper om de flesta teknikområden i databranschen: operativsystem, programmering, databaser, modellering osv. Men också mer övergripande kunskaper om teknikens tillämpning, liksom erfarenheter av både forskning och praktik.

-Det är viktigt att ha ett inflöde både från den akademiska världen och från industrin, säger Lars-Åke Johansson. Det gäller både att kunna konstruera och att begripa vad grejorna kan användas till. Man måste förstå sig på industrins verklighet och kunna diskutera på deras villkor.

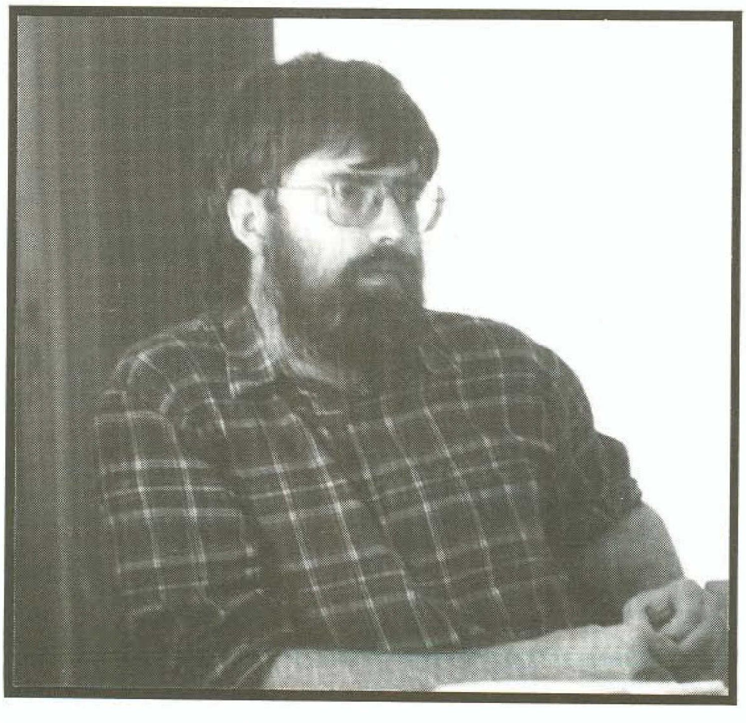
En av de nyare på CASE-gruppen är Per Bergsten, som började på SISU i september. Han arbetade tidigare som konstruktör på en konsultfirma som specialiserat sig på Unix-system, men har också erfarenhet av forskarvärlden. Han uppskattar det långsiktiga i SISUs verksamhet.

-Som konsult arbetar man oftast med mindre och avgränsade uppdrag, säger han. Här handlar det mer om långsiktig utveckling, med projekt som löper över flera år. Det är intressant att syssla med problemen i företagens verksamhet, i produktionen.

Varierad miljö

Personalen på Göteborgskontoret tycks trivas. Det är roligt att jobba med CASE, säger många. Säkert bidrar blandningen av forskning och praktik till trivseln. Omväxling förnöjer, som bekant.

Även vad gäller maskiner finns det möjlighet till omväxling. På kontoret finns arbetsstationer från Sun, Digital och IBM. Ramatic utvecklas på alla fabrikaten, och intressentföretag som vill veta mer om arbetsstationer och CASE-verktyg är välkomna på besök. Dessutom används Ramatic som testbänk för framtida egenskaper hos CASE-verktyg. Erfarenheterna från tillämpningsprojekten ger impulser till nya användningsområden och tillför SISU kunskap om vilka behov intressenterna har.

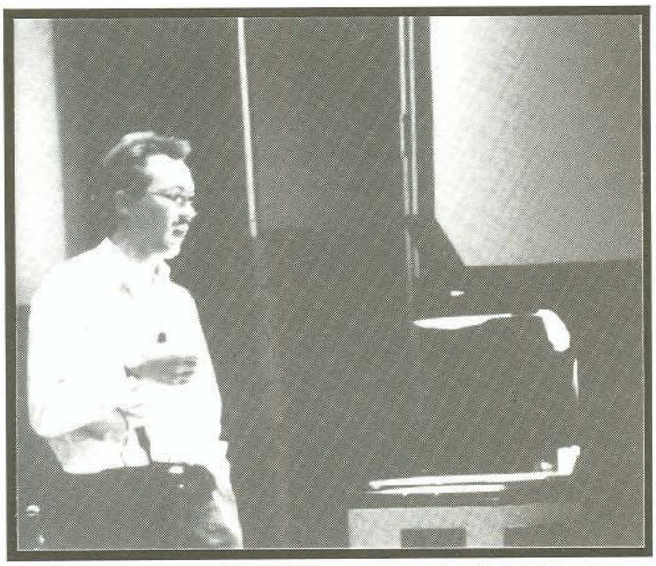


Per Bergsten

How to Test and Compare CASE tools

Tapani J. Kinnula,
Jalal Matini,
Swedish Institute for
Systems Development

Vid CASE89-konferensen presenterades ett föredrag av Tapani Kinnula, vilket här återges i oavkortad form. I SISU Rapport nr 3 redovisas det arbete som ligger till grund för denna artikel.



Tapani Kinnula "in action" på CASE89.

This paper describes how a practical study of computerized tools for systems development could be prepared and made. The paper suggests the use of a detailed checklist combined with a description how the practical evaluation of CASE tools should be executed. The purpose is to ensure that the evaluations of different CASE tools will be made in similar ways and that results can easily be compared when selecting a strategy for computer aided software engineering.

Introduction

The complexity of design of computer systems is increasing. Practically all efficient corporations are moving to a higher level of automation in their data processing. This implies higher requirements on the computer systems being developed. Consequently, the need for more efficient methods and tools for developing and maintaining computer systems is increasing. During the last few years, a large number of tools for analysis and design of computer systems have been born to daylight. These tools, called CASE tools, intend to automate parts of the systems development process. There is, however, a great variety among CASE tools today. Some of them are merely graphic editors with a simple database for saving the design information, often can these specifications be transferred to 4GL-tools or database systems for further design and implementation. Some tools, in turn, are sophisticated workbenches with a large amount of automation in analysis and design processes and with possibility of generating code from design specifications.

An important factor related to CASE tools is the method (or methods) which the tool supports in systems analysis and design. An enterprise may have to decide whether to invest in a method-specific CASE tool and learn the method it use, or whether to choose a tool that can be customized to support the method used by the enterprise. To make this important decision, detailed information about the tool and supported methods is needed. Further requirements for specific functions, as prototyping, code genereration and interfaces to other products, must be considered when choosing a tool. Of course, there are other aspects too when selecting a CASE tool, but they are beyond the scope of this paper. The purpose of this paper is to describe how reliable and detailed information about CASE tools can be captured by studying and assessing them in practice.

Justification

There should be no doubt that the best way to learn a CASE-tool is to use it in practice. Reading information from vendors and developers may form the basis of preliminary selection, that is, make it clear what tools do not satisfy the basic requirements. However, to choose the right tool, detailed information is needed about several tools, their advantages and the problems that can appear when using them in practice. Putting a CASE tool into practice is an efficient way to discover the practical problems and how serious they are. In particular, when studying several tools in practice in similar ways, important experiences for comparing the practical values of these tools are obtained.

The purpose of a practical study is to obtain experiences and relevant features of the studied object and to formalize the results. The resulting report forms a basis of the final assesment of the studied object. When studying CASE tools, the report should provide information about all important tool features and their role in practice. Detailed and identically structured reports on several tools provide a good basis for comparing tools and making the final selection.

Of course, preparing and executing these practical studies of CASE tools is an expensive and time consuming task. Many enterprises are confined to put up with vendor information and experiences from other corporations, when making their decision. User experiences are, however, seldom written down and formalized in order to make them easily accessible for other people. Therefore, well formed reports on practical studies and evaluations of CASE tools provides an important way of getting valuable and detailed information about the tools from the practical point of view.

To make studies of different CASE tools and resulting report as equal as possible an inspection protocol could be used. An Inspection protocol is a kind of check list of tool attributes including the description of how these attributes should be interpreted and studied. Using the protocol during the practical test work and report writing ensures well structured and comparable reports that can be used in the selection process. Figure 1. illustrates our approach for testing and comparing CASE tools.

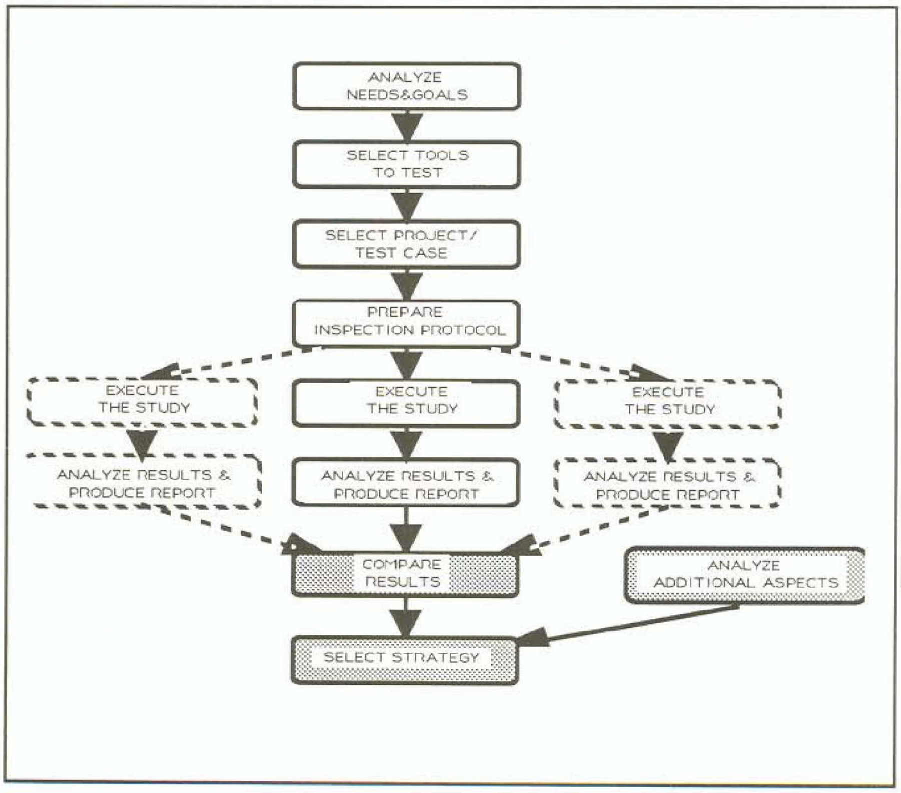

ANALYZE NEEDS&GOALS
SELECT TOOLS TO TEST
SELECT PROJECT/ TEST CASE
PREPARE INSPECTION PROTOCOL
EXECUTE THE STUDY
EXECUTE THE STUDY
EXECUTE THE STUDY
ANALYZE RESULTS & PRODUCE REPORT
ANALYZE RESULTS & PRODUCE REPORT
ANALYZE RESULTS & PRODUCE REPORT
COMPARE RESULTS
ANALYZE ADDITIONAL ASPECTS
SELECT STRATEGY

Figure 1. Testing and comparing CASE tools. Several tools can be tested simultaneously and the resulting reports form a basis for selection of strategy for CASE technology. Note, that the grey activities indicate activities of the enterprise, while the white activities could be performed by independent institutes or other enterprises as well.

Preparing a Practical Study

Before getting into work and studying CASE tools, some preliminary work should be done. The first thing is to identify needs and goals: what are the problems with the current systems development environment? How to solve these problems? what goals do we want to achieve, now and in future? When the needs and the goals are known it is easier to decide which tool type is suitable, and how to form tool studies.

When studying a CASE tool, it is necessary to know what features and lacks to look for. The requirements for the tool should be analyzed and the related tool features should be consedered when forming the inspection protocol and applying it to the tool. This ensures that no relevant aspects will be omitted during the practical study The protocol also provides a well-defined structure for the resulting report.

Identifying Needs and Goals

To be able to decide what to do, one has to know what's wrong. The most usual reasons to buy a CASE tool concern productivity in systems development and the quality in resulting systems. Knowledge about why the productity is too low and why the systems quality is not good enough is necessary when selecting a strategy for CASE. For example, implementing a method with many weak points in a CASE shell will not succeed without profound and costly improvements of

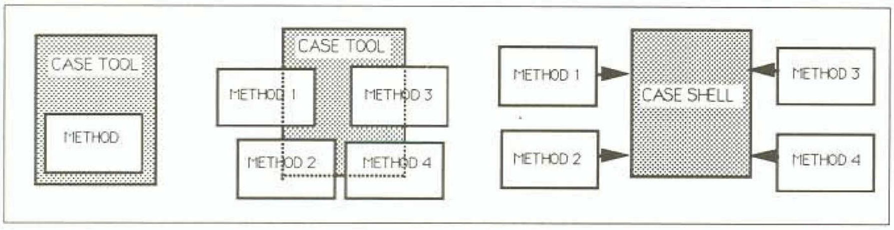


Fig 2. Different types of CASE tools. a) method-specific CASE tools have built-in methods, b) a CASE tools supporting several methods, but none of them completely, c) a CASE shell has no built-in method support, but allows implemention of different methods and method specific techniques

the method. In the following we present some questions of concern when analyzing needs and goals.

- **what are the current problems**
- **when and how does the problems appear**
- **what are the basic causes for the problems**
- **what are the consequences of the problems**
- **how can these problems be eliminated**
- **how can a CASE tool solve the problems**
- **what else do we want to achieve with a CASE tool**
- **what kind of CASE tool is suitable**
- **what are the basic demands for the tool**
- **what methods should the tool support**
- **what specific features are desired currently**
- **what features are needed for future requirements**

Answers to these questions provide a foundation for a preliminary evaluation ofCASE tools. Tools not satisfying the basic requirements can be detected and omitted from further consideration. There are lots of brief analyses and surveys of commercially available CASE tools that can be helpful in this stage.

Analysis of current problems may result in discovery of serious lacks in the currently used method, which indicates a need of a more efficient and formalized method. Which metod to choose depends on various things, but it should be pointed out that methods provided by specific tools often have a great priority. Some tools have strong connections to well-defined and formalized methods and therefore support a great deal of automation in different phases of systems development. It may be appropiate to examine tools in this class when new and efficient methods are needed. On the other hand, if the method used earlier turns out to be efficient and well defined, it is convenient to examine the possibilities of implementing it within a CASE shell.

Problems or special needs that do not have their origin in the method or that are related to certain kind of applications should also be taken into consideration when selecting and reducing the set of potential tools.

Evaluating CASE tools in real projects requires both financial and time resources, but selecting the right tool, and even more important, the right method, may result in considerable savings in productivity and systems quality.

Test Case or Real Case?

There are to possibilities to evaluate a CASE tool in practice: a more or less ideal test case can be used, or the tool can be tested within a real systems development project. Which strategy to choose depends on available time and resources. It might be of advantage for an enterprise to have well structured reports based on test cases at hand when they need to make a quick decision or do not have the necessary resources to test tools within real projects. However, most enterprises cannot test a number of tools by themselves, no matter whether using a test case or a real case.

Test cases seem to be most convinient to be used by consulting enterprises or institutes that are specialized in testing tools for systems development in order to publish the results. Test cases also serve the simplest way of testing tools and tool prototypes for researh purpose. They can be especially formed to test specific features like expressibility (that is, how many types of details and constraints can be expressed by using a tool's description techniques).

Testing tools with synthetic test cases is easier and can be carried out at any time, by contrast with real projects. Putting tools on real and heavy cases is however the most reliable way of getting information about the practical value of tools and methods when used in a specific environment. An other aspect that argues against test cases is that they are already conceptualized: conceptualization of the physical and logical environments is one of the main problems in systems development- and one of the main reasons to use CASE tools, while the complete "case description" for a test case is nothing else than a conceptualization of a virtual environment.

Evaluating CASE tools in real projects requires both financial and time resources, but selecting the right tool, and even more important, the right method, may result in considerable savings in productivity and systems quality. Therefore, enterprices with many heavy projects and complex computer systems probably serve their purposes best if they choose to put the most potential tools on real systems development projects when evaluating them in practice.

When using a synthetic test case, some important aspects should be taken into consideration. A good test case is not too comprehensive in time and effort while being characteristic and wide enough to cover the most important features on a specific area. What test case to select (or design) depends on the type of subject area and what features and problems to examine. The test case ought to be well-defined and described in great detail to allow people that do not have possibilites to interview the "real users" to be able to implement it in reasonable time. Unclear and ambiquous descriptions should be avoided. In real projects much time and effort are spended to solve problems with ambigous and unclear descriptions and modelling different kind of solutions to these problems. It is not relevant to burden a test case with these problems, because the are not "productive" enough with respect to consumed time and obtained results.

The target environment of the system and the system type should be reflected by the test case. For example, a test case specifying a database-oriented information system is not particularly suitable when testing tools for real-time control systems. The choice of the test case is of great importance in order to detect the right features in the right environment.

It is often possible to choose an existing test case and use it with only slight modifications. There are several test cases that are frequently used when testing and evaluating methods for systems development. These test cases are most often appropiate also for testing and evaluating tools for systems development. Of course, it happens that a new test case must be designed from the beginning, but it is probable that there are "real cases" that can be simplified and modified without too much effort.

Inspection Protocol

The inspection protocol presents and treats all relevant aspects of concern when studying a CASE tool. These aspects are normally related to tool attributes and how they should be treated when evaluating the tool or writing the report. The purpose is to provide a uniform and well-defined way of executing the study and equal treatment of tools in the resulting reports. Reports based on equal inspection protocolls have equal structure and organisation which make it easy to assess tools and compare them with each other. More specially, the features of particular interest, when making the final selection of the tool, can easily be picked upp and examined in detail. A carefully accomplished execution of the protocol also provides a means of indicating the weak points of the studied tools which helps developers to find out possible improvements to be carried out in future releases.

We have emphasized that the test case should reflect the target environment and particular problems of interest. That is valid for the protocol too. The protocol should reflect the tool type and treat corresponding features. In other words, the protocol is not general in the common sense, it is more or less tailored for different types of CASE tools. Of course, one could use a general and detailed

The inspection protocol

A. Presentation

B. Methods and techniques

B.1 Coverage
B.2 Techniques and diagramming tools
B.3 Integration between diagramming tools
B.4 Method support

C. Database

C.1 Data dictionary
C.2 Data access and consistency in data representations
C.3 Analysis functions, reports and error discovery
C.4 Data security

D. User environment

D.1 User interface and user interaction
D.2 Help
D.3 Multiuser environment

E. Features and facilities

E.1 Portability
E.2 Communication
E.3 Transformation of data structures
E.4 Code generation
E.5 Prototyping
E.6 Reusability of code and specifications
E.7 Flexibility
E.8 Information control
E.9 Project management support

F. Effects on the process of systems development

F.1 Productivity and quality improvements
F.2 Effects on systems life cycle
F.3 Role of end users
F.4 Education

G. Hardware, software, documentation and costs

H. Conclusive remarks

H.1 Tool summary and assessment
H.2 Future improvements
H.3 Subjective impressions

protocol valid for all kinds of tools and only treat the aspects that are relevant for the current tool.

In the following text we present the protocol we used when studying CASE tools DEFT™ and IEW™. The protocol is formed for database-oriented informations systems and does not treat aspects like those of control systems or real-time applications support. The protocol is not especially appropiate for studying CASE shells either. However, it can easily be augmented with new type of aspects and attributes. Note that the tools attributes are not always given explicitly, but as questions or in descriptive way in order to avoid ambiquities and unnecessary confusion.

The inspection protocol

Terms like 'diagram', 'representation technique' and 'description technique' are usual in the following text. Normally, 'representation techniques' and 'description technique' include the term 'diagram', that is they are more general and include also different kinds of textual representations. For simplicity, however, we often use the term 'diagram' for all kinds of description techniques.

A. Presentation

- Give a brief description of the tool
- developer and vendors
- environments (PC, Macintosh, Mainframe, workstations, networks)
- tool architecture (i.e. the modules/components that the tool consists of and main task for each of them)
- tool type : method-specific, CASE shell, no specific methods

Note: no details here, just a brief orientation about the tool

B. Methods and techniques

B.1 Coverage

- What phases and stages of the systems devlopment process are covered by the tool?
- Motivate and exemplify your conclusion about the coverage (schematic figures may be very illustrative)

Note: most commonly used division: planning, analysis and specification, design, implementation, testing, installing and maintenance)

B.2 Techniques and diagramming tools

- Describe and explain the existing diagramming or description techniques
  - main usage
  - syntax and constraints (e.g. JSP syntax)
  - how syntax and constraint checks performed are performed
  - power/expressiviness of the description techniques
  - customizability (e.g modification of drawing symbols or augmenting the tool with new symbols)
- Illustrate with figures and examples of diagrams and suchlike

B.3 Integration between diagramming tools

- Describe and explain different types of connections between the diagrams
- Are the connections dynamic, that is, does a modification of a diagram imply related modifications in other diagrams?
- How useful are the connections in practice? (e.g you need not to specify objects more than once)
- can new diagrams be opened from the current diagram (e.g to allow more detailed specifications of a design object)?
- Can specifications be transferred (shared) between different types of diagrams or representations?
- Does the tool support automatic redrawing of the transferred (or shared) objects?

B.4 Method support

- What methods does the tool support?
- If the tool is customizable, what methods and techniques are supported or can be choosen on delivery?
- How method-dependent is the tool, or in other words, is the tool dedicated to a particular method (or methods) and does not allow any departures from the method?
- The systems development process is often divided into phases, and different methods often support different phases. If possible, describe how the supported methods and description techniques are related the the phases of systems devlopment process
- What are the consequences of the degree of method dependence? For example, a strong method dependence may limit the user's possibilites to express or model specific features and constraints. On the other hand, a strong method dependence may force the users to a formalized and efficient way of working and therefore indirectly improve productivity and systems quality.
- Does the tool dictate or constrain the way of using the provided description techniques (e.g. to allow proper analyses or efficient code generation)?
- Are accessible analysis and checking functions related to the selected method?

C. Database

When talking about a tool's underlying database a few aspects should be pointed out.

Firstly, the database is the heart of the tool, almost all actions, as creating, modifying and deleting objects and analyzing the relationships between dem, are in some way influenced by the database and its capacity.

Secondly, diagramming tools are most often just an interface to the database, and gives us a simple and efficient way of seeing and manipulating the objects stored in the database. Therefore, the interaction between the database and description techniques is of great importance.

Thirdly, the capacity and usefulness of available analyses and controls is due to the contents of the database and how it is managed.

C.1 Data dictionary

We use the term 'data dictionary', but 'repository' or 'encyclopedia' would do as well!

- Database type (relational, hierarchical, etc.)
- What is stored in the data dictionary and what are the associations between stored data? Try to draw a conceptual schema on the database contents, if possible.
- Is the dictionary uppdated online or by command whenediting in diagrams?
- Can the data dictionary contents be imported from and exported to other tools?
- Can dictionaries be split and combined (e.g subproject <-> main project)?

C.2 Data access and consistency in data representations

- Is the dictionary contents immediately accessible for viewing and editing (e.g through a special dictionary editor)?
- Do all modifications made in diagrams imply uppdating of dictionary contents (and vice versa)?
- Is the data dictionary always consistent with diagrams?
- Are the diagrams always consistent with each other? (Note: This aspect is partially treated in section B.3)

C.3 Analysis functions, reports and error discovery

Analysis functions, reports and error discovery are often related to specific types of diagrams. It might therefore be a good idea to treat these functions from the diagrams' point of view.

Since reports are an important part of systems documentation and form the basis of manual analysis, the tool should generate well organized and relevant reports. Unnecessary or inappropiate information should be eliminated by the tool or by user options.

- What kinds of analyses/error-checks are performed on-line (e.g. syntax checks or warnings for inconsistencies when the user tries to perform inappropiate actions?)
- What other kind of analyses and checks does the tool support? (some desirable analyses: find inconsequences between diagrams, find objects that are not used or are not defined, find stand-alone objects, i.e. objects that have no associations to other objects)
- Examine the report functions. Is the user reduced to make use of existing reports? Can the user customize them or define his own reports?

C.4 Data security

- What kind of security functions does the tool support? Some examples:
  - auto-save (memory -> disk)
  - back-up function (automatic or by command)
  - dictionary 'recovery' or 'repair' functions
  - data access security: passwords, user privilegies,
  - data uppdate control in multiuser/network environments

D. User environment

D.1 User interface and user interaction

- Describe briefly the user interface (windows, mouse-support, pull-down/pop-up menues, etc.)
- Describe briefly the user interaction, that is how the user communicates with the tool and vice versa (Examples: User enters data through a command-line or a dialog box, draws and moves objects with mouse, the tool shows messages on a message-line or in a message box appearing on the screen).
- How many diagrams or windows can be opened and elaborated simultaneously?
- Does the tool support the user in navigating between diagrams or between hierachies of specifications?
- Does the tool warn the user when performing dangerous actions?
- Can the user "undo" actions

Illustrative figures might help!

D.2 Help

- Describe the available help functions:
  - on-line help
  - context sensitive help (i.e. right help in right situation)
- How detailed is the help information
- How 'intelligent' help is available (e.g. the tool suggests some possible actions to take in a certain situation)

D.3 Multiuser environment

If the tool supports networks or multiuser environments, describe how this influences the situation of a user.

How to Test and Compare CASE tools

E. Features and facilities

E.1 Portability

- Can specifications be transferred to other tools/environments

E.2 Communication

- Examine communication capabilites
  - built-in communication facilities
  - communication with DBMS, 4GL-systems, other systems
  - network support

E.3 Transformation of data structures

- Can specifications (e.g. Entity-Relationsship diagrams, data structure diagrams etc.) be transformed into database schemas of a DBMS.
- Can database schemas be transferred back to the CASE environment and redrawn?

E.4 Code generation

- examine the code generating facilities. Some points of interest:
  - code type (pseudo,4GL, COBOL,C,etc or compiled)
  - target environments
  - completeness of code: executable code, skeleton programs or declarations of datastructures only
- how powerful/easy-to-use is the coding facility?

E.5 Prototyping

- what kind of prototyping facilities are supported?
  - simulation of user dialogues
  - execution/interpretation of specifications or state-transition diagrams
- Does the tool support implementation of small 'test systems' which can be tested and completed gradually?

Note: powerful coding facilities eventually combined with reverse engineering might do as
a kind of prototyping

E.6 Reusability of code and specifications

- examine reusability of object definitions and design specifications (including code).
  Some points of interest:
  - reuseability, both in the current and in future projects
  - libraries for saving specifications to be used in other contexts
  - support for searching among saved packages or libraries
- Reusing old specifications puts reguirements on analysis and error detecting functions. It might be an good idea to comment these functions from this specific point of view.

E.7 Flexibility

- Describe how the tool can be customized or extended.

E.8 Information control

When talking about information control, three types of control facilites can be considered and treated:

- Software configuration control: when the tool environment can be customized to user needs or different methods/ techniques, the issue of compatibility between different tool customizations arises!
- Version/release control: compatibility between old and new versions of the tool.
- Information change tracking: detecting and managing alternative or different old versions of system specifications.

E.9 Project management support

- Does the tool provide project managing facilites?
- Can the tool be integrated with separate products for project management

F. Effects on the process of systems development

F.1 Productivity and quality improvements

Improvements in productivity and quality are often hard to estimate, since they depend - in addition to the tool- also on the lentght of the test period, the way of working, methods used, education in the method and tools. However, some well motivated approximations are desirable. We list some effects to consider when discussing these aspects:

- effects deriving from improvements in methods and organisation (often hard to estimate)
- better co-ordination of the project due to central dictionary
- more effective human communication due to illustrative and precise diagrams and automatic documentation
- improvements directly derivable from the tool capabilities (time savings, documentation, etc.)

F.2 Effects on systems life cycle

- What phases in systems life cycle does the tool act on and change?
- What steps in these phases are affected?
- How striking are the effects?
- What tool features are these effects due to?

F.3 Role of end users

- How can end-users be involved in the development process when using the tool?
- How does the end-user involvement affect the resulting system and its quality?

F.4 Education

Education is necessary to succesfully introduce a CASE tool and new methods in an enterprise. Later on, it is important to keep the knowledge up to date to ensure optimal use of the tool and methods. The following aspects should therefore be discussed.

- What are the desirable user qualifications?
- what courses on the tool and its methods does the vendor provide?
- What level of knowledge do these courses focus on (overview, more detailed, expert knowledge)?
- Is the tool/method of such complexity that it is advisable to have persons with 'expert knowledge' available in the enterprise?
- Considering the previous question, is there any risk for becoming dependent on external knowledge due to the complexity of the tool?

G. Hardware, software, documentation and costs

- Describe the hardware and software configurations in detail.
- What are the costs of hardware, software and education.
- Describe the documentation and its quality.

H. Conclusive remarks

H.1 Tool summary and assessment

- Summarize the tool and its disposition in the process of systems development
- What kind of enterprises/users can make the best use of the tool
- discuss advantages and disadvantages of the tool, but keep in mind that such things are often related to specific needs and requirements (i.e. to a specific enterprise).
- If the resulting report address a particular enterprise, discuss the value of the tool for that enterprise

H.2 Future improvements

- what improvements and changes will be carried out in future releases
- what improvements should be carried out? Motivate!

H.3 Subjective impressions

- Here you can tell about your private impressions.

Executing the Study

To carry trough the study without problems, it is advisable to examine the educational needs before strarting the study. It will certainly take a much longer time to learn a tool and its methods through using them in practice (trial-and-error-method) than a vendor course would take. Furthermore, good education is necessary when putting the tool on a real project, otherwise the project (and the study) will not be succesful.

Though not necessary, it as a good idea to get information about similar and commercially avalaible tools too. A survey of the state of the art in CASE area might help when assessing capabilities of a tool in proportion to other tools. However, vendor prospects are often fancy and have a greater bent for telling what the tool can do while omitting what it cannot do. To get more relevant and detailed information, it might be helpful to make use of the inspection protocol. The protocol serves as a checklist and a source of ideas when acquiring information about other tools.

Implementing the test case should be carried out as any systems development project. Of course, since the number of persons involved is probably small, no heavy project administration is necessary. The main exception is the protocol. When applying the tool and new methods, it is an good idea to keep the protocol in mind and write down experiences continuously. The protocol is helpful when formalizing experiences, and even more important, it gives ideas of what features and lacks to look for. Having the protocol as a guide and a notebook when examining the tool also reduces the risk of not taking some important aspects into consideration.

When writing the report, the organisation of the protocol should be followed, since it makes it easy to formalize and structure the obtained experiences. Furthermore, a standardized report to be used when comparing tools is one of the main advantages of the use of an inspection protocol. Knowledge about other tools and methods helps you to form a balanced and objective description of the the tool. It also makes it easier to analyze the problems that arose during the implementation work; whether they were due to shortcomings of the tool or insufficient knowledge.

Making the Decision

Although our focus is on the the practical study and the inspection protocol and not on how the final decision on investments in CASE technology should be made, we present some aspects worth considering.

- how well the tool satisfies the current and future requirements
- current hardware/software environment
- plans for future hardware/software environment
- future development of CASE tools (research results)
- changes needed in the organisation (new tools, new methods, new responsibilities)
- educational needs and investments

Additional aspects besides these and which strategy to choose are discussed more detailed in [Bubenko88].

Summary and Concluding Remarks

We have presented a way of testing CASE tools in practice. The idea is that detailed and relevant information is obtained about a CASE tool through putting it on a sythetized test case, alternatively a real systems development project, and analyzing its capabilities and features with assistance of an inspection protocol. The protocol serves as a checklist of tools attributes and what to examine including explanations and justifications of a variety of tools features. The protocol is supposed to be used both as an inspection guide during the practical work and as a skeleton for resulting reports. This ensures more equal studies and standardized reports that provide detailed and reliable information about tools and methods and form a basis for precise comparisons between tools.

This paper is mostly based on our work on studying commercially available CASE tools and our experiences from practical studies. We conclude with some practical advises worth considering when making practical studies on CASE tool.

- prepare carefully: analyze your needs and expectations, choose right tool type and test case (if you are going to use one)
- examine and complete your protocol with desired features and explanations
- have a good training in both tools and methods before you start, discuss with experienced users if possible
- make detailed notes during the practical work, use the protocol as a guide
- analyze your experiences and discuss the problems and the protocol with other users, compare your experiences
- compare with other similar tools, you might get some ideas
- check everything that is unclear with vendors or developers to avoid missunderstandings
- let the vendor or developers comment the report, you might have misunderstood or missed something, and moreover your critics might give them ideas of future improvements

REFERENCES

Bubenko88
Janis A. Bubenko jr, Selectin a Strategy for Computer Aided Software Engineering (CASE), SYSLAB, 1988

Brodie87
M.L. Brodie, Automating Database Design and Development,Tutorial, Nov 1987, Authors address: GTE Laboratories Inc, 40 Sylvan Road, Waltham, MA, 02254, USA

Floyd86
C. Floyd, A comparative Evaluation of Systems Development Methods, in [Olle86]

Martin88
C.F. Martin, Second Generation CASE Tools: a Challenge to Vendors, IEEE Software, 1988

Olle86
T.W. Olle, H.G. Sol, C.J. Tully (editors), Informations System Design Methodologies: Improving the Practice, Elseview Science Publishers B.V. (North Holland), Amsterdam, 1986

Pressman88
Roger S. Pressman, Making Software Engineering Happen, Prentice Hall, 1988

Rock89
Rosemary Rock-Evans, CASE Analyst Workbenches: a Detailed Evaluation, Ovum Ltd, 1989

Starts87
The Starts Guide: a guide to methods and software tools for construction of large real-time systems, NCC Publications, 1987

SISU PUBLIKATIONER - BESTÄLLNING

SISU Analys och SISU Rapport är skriftserier som tas fram inom SISUs ramprogram och finansieras av intressenterna. Därför håller vi mycket låga priser för att nå spridning inom intressentkretsen.

BESTÄLLNING till

SISU
Box 1250
164 28 Kista

Kolla gärna med din kontaktperson. Hon eller han får i regel en dubbel uppsättning av analys och rapport.

Upplysningar i övrigt kan du få av Marianne Sindler eller Lars Bergman, SISU, 08 - 752 16 00.

Härmed beställs publikationer enligt följande:

□ ex Analys nr 1: Konceptuell Modellering (1985)

□ ex Analys nr 2: Några aspekter på kontorsinformationssystem (1985)

□ ex Analys nr 3: Grafiskt baserade datorstöd för systembeskrivning (1986)

□ ex Analys nr 4: ADA-teknologi (1986)

□ ex Analys nr 5: Databaser - enkla att hantera (1987)

□ ex Analys nr 6: An Introduction to Distributed Database Systems (1987)

□ ex Analys nr 7: Kunskapssystem (1988)

□ ex Analys nr 8: OSI (1988)

□ ex Analys nr 9: Meddelandehanteringssystem (1988)

Medlemspriset är 100 SEK per nummer av Analys

□ ex av CASE89-föredragen, ca 700 sidor (begränsad upplaga).
Priset är 675 SEK för medlem i ISVI-organisation.

□ ex Rapport nr 1/87: Ett förslag till referensmodell för Människa-Dator-interaktion
Priset för Rapport nr 1/87 är för medlem i ISVI-organisation 50 kr.

□ ex Rapport nr 2/87: Generellt teknikstöd för ärendehantering
Priset för Rapport nr 2/87 är för medlem i ISVI-organisation 100 kr.

□ ex Rapport nr 3/88: En experimentell studie av CASE-verktygen Deft och IEW/WS
Priset för Rapport nr 3/88 är för medlem i ISVI-organisation 150 kr.

□ ex Rapport nr 4/88: RAMATIC på Volvo Personvagnar.
Priset för Rapport nr 4/88 är för medlem i ISVI-organisation 80 kr.

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

Namn	
Befattning	
Avdelning	
Företag/organisation	
Adress	
Postnr-Adress	
Telefon	

SISU-matrikel

ARTHUR YOUNG AB
Anders Norén
Box 3143, 103 62 Stockholm
Tel: 08/796 33 00

ABB DATA AB
Gunnar Nilsson
721 80 Västerås
Tel. 021/32 33 00

AU-GRUPPEN AB
Sven-Bertil Wallin
Kungsg. 53, 111 22 Stockholm
Tel: 08/24 34 20

DATA LOGIC AB
Örjan Odelhög
Fröfästeg. 125, 421 31 Västra Frölunda
Tel: 031/45 03 40

DIGITAL EQUIPMENT AB
Staffan Westbeck
Allen 6, 172 89 Sundbyberg
Tel: 08/733 80 00

ENEA DATA SVENSKA AB
Bo Steinholtz
Box 232, 183 23 Täby
Tel: 08/792 25 00

ERICSSON
Christer Dahlgren
IT-plan, Mäster Samuelsg 10, 111 44
Stockholm
Tel: 08/24 59 80

FÖRSVARETS MATERIELVERK
Barbara Hedlund
FMV, Elektro LT
115 88 Stockholm
Tel: 08/782 67 35

FÖRSVARETS RATIONALISERINGSINSTITUT
Mikael Franzén
FRI, Box 80008, 104 50 Stockholm
Tel: 08/788 75 00

FÖRSVARSSTABEN
Rudi Olsson
Box 80001, 104 50 Stockholm
Tel: 08/788 75 00

IBM SVENSKA AB
Lars Arosenius
163 92 Stockholm
Tel: 08/793 40 60

INFOLOGICS
Dick Eriksson
SU TVT Infologics AB,
Chalmers Teknikpark,
412 88 Göteborg
Tel: 031/72 42 60

IRM CONSULT AB
Eskil Swende,
Smedslättens gård,
Skogsfrugränd 21, 161 38 Bromma
Tel: 08/80 22 40

KOMMUNDATA AB
Gunnar Sandberg
125 86 Älvsjö
Tel: 08/749 80 00

MIMER SOFTWARE AB
Lars-Erik Jansson
Box 1713, 751 47 Uppsala
Tel: 018/18 50 00

PEAB
Stellan Borg
Philips Elektronikindustrier AB
Dream, 175 88 Järfälla
Tel: 08/703 10 00

POSTEN
Gert Persson
Koncernstab utveckling,
105 00 Stockholm
Tel: 08/781 10 00

AB PROGRAMATOR
Håkan Friberg o Per Tidén
Box 20072, 161 20 Bromma
Tel: 08/799 35 00

RIKSSKATTEVERKET
Carl-Göran Svensson,
171 94 Solna
Tel: 08/764 88 74

SAAB-SCANIA
Sven Yngvell
Flygdivisionen Dataservice
581 88 Linköping
Tel: 013/18 23 86

SAS DATA
Ove Lundvall
161 87 Stockholm
Tel: 08/797 10 18

S-E-BANKEN
Larsaxel Johansson
SEB Data/Metoder H3
106 40 Stockholm
Tel: 08/763 50 00

SKANDIA
Anders Fungdal
Skandia-Data, 103 50 Stockholm
Tel: 08/788 17 26

SKF
Bo Lindahl
SKF Group Headquarters
415 50 Göteborg
Tel: 031/372626

SPADAB
Göran Lustig
Box 341, 101 24 Stockholm
Tel: 08/13 41 54

STATSKONSULT ADMINISTRATIV UTVECKLING AB
Per-Olof Hultman
Box 4040, 171 04 Solna
Tel: 08/730 03 00

STATSKONTORET
Kerstin Norrby, Lars Hellberg
Box 34107, 100 26 Stockholm
Tel: 08/738 45 94, 08/738 47 77

SÖDRA SKOGSÄGARNA
Jerry Nilsson
Södra Data AB,
Box 832, 264 00 Klippan
Tel: 0435/12090

TELEVERKET
Henry Samuelson
ADB-Service
Box 164, 136 23 Haninge
Tel: 08/707 10 00

TELEVERKET
Avd f Grundteknik
Birgitta Carlsson
123 86 Farsta
Tel: 08/ 713 38 51

Karl-Erik Carlsson
Telelogic AB
Box 883, 851 24 Sundsvall
Tel: 060/16 14 44

TELUB TEKNIK AB
Hans Holmberg
Box 1381, 171 27 Solna
Tel: 08/730 50 00

UNISYS AB
Inge Dahlberg
171 91 Solna
Tel: 08/55 15 00

VATTENFALL
Helge Holmén
Statens Vattenfallsverk,
Vattenfall Data, ADS, 162 87 Vällingby
Tel: 08/739 50 00

AB VOLVO-DATA
Kenneth Pettersson o Anders Persson
Avd 2800, 405 08 Göteborg
Tel: 031/66 76 48,66 56 48

VOLVO LASTVAGNAR AB
Magne Källström ,
Avd 24180 BC4, 405 08 Göteborg
Tel: 031/ 66 52 50

VOLVO PERSONVAGNAR AB
Uno Eriksson
Avd. 50820 AU, 405 08 Göteborg
Tel: 031/592074

VÄGVERKET
Björn Oresand
781 87 Borlänge
Tel: 0243/750 00

ÅF MANDATOR AB
Ingemar Dahlgren
117 81 Stockholm
Tel: 08/709 31 73