

Föredrag på konferensen "Systemutveckling i praktisk belysning"
19/11 1991 i Norrköping, Dataföreningen - Östra kretsen

STÖD OCH STRUKTUR I SYSTEMUTVECKLINGSPROCESSEN

av

Göran Goldkuhl
FoU-gruppen VITS
Institutionen för datavetenskap
Universitetet i Linköping
581 83 Linköping
tel 013-28 14 52

Sammanfattning: Vid utveckling av datorbaserade informations-system behövs olika hjälpmedel. Man använder ofta modeller, metoder och datorverktyg. Rapporten ger en referensram över dessa hjälpmedel och hur de kan samspela i systemutvecklingsprocessen. Olika historiska utvecklingstendenser går igenom:

- Olika utvecklingsmetoders framväxt
 - Ökad användning av verktyg
 - Konsultmodeller som kombinerar olika metoder
- Rapporten innehåller också en diskussion om några tänkbara framtida utvecklingstendenser:
- Synsättens ökande betydelse
 - Pendling mellan standardisering och innovation
 - Situationsanpassning av metoder
 - Egenkombinerade metodpaket
 - Anpassning av CASE-verktyg

Denna rapport har utarbetats inom ramen för projektet PRIMCASE (principer för metodanpassning av CASE-verktyg) som bedrivs med stöd från NUTEK/ITYP.

INNEHÅLL

	Sida	
1	EN REFERENSRAM	1
1.1	Bakgrund	1
1.2	Några begrepp	1
1.21	Modell och modellstruktur	1
1.22	Metod	2
1.23	Verktyg	4
1.24	Synsätt	5
1.3	En begreppsstruktur	6
2	NÅGRA HISTORISKA UTVECKLINGSTENDENSER	11
2.1	Olika utvecklingsmetoder	11
2.2	Ökad användning av verktyg	13
2.3	Konsultmodeller som kombinerar olika metoder	15
3	NÅGRA FRAMTIDA UTVECKLINGSTENDENSER	17
3.1	Synsättens ökande betydelse	17
3.2	Pendling mellan standardisering och innovation	18
3.3	Situationsanpassning av metoder	19
3.4	Egenkombinerade metodpaket	20
3.5	Anpassning av CASE-verktyg	21
3.6	Avslutning	23
	LITTERATURREFERENSER	24

1 EN REFERENSRAM

1.1 BAKGRUND

Det talas mycket om modeller, metoder och verktyg för systemutveckling. Vilken betydelse har de för utvecklingsarbetet? Hur hänger modell, metod och verktyg ihop? Vad menas med modell, metod och verktyg? Vad är det för skillnad mellan modell och metod?

Detta är några frågeställningar som jag avser att behandla i denna rapport. Jag ska försöka ge en innebördsbestämning av de viktiga begreppen modell, metod och verktyg. Det handlar om begrepp och företeelser som finns för att ge struktur åt och stöd i systemutvecklingsprocessen. Jag kommer också följa några utvecklingstendenser i en bakåtblick och framåtblick.

1.2 NÅGRA BEGREPP

1.2.1 Modell och modellstruktur

Modell, metod och verktyg är begrepp som används för att tala om företeelser inom området systemutveckling (SU). Med systemutvecklingsmodell brukar ofta avses en övergripande struktur för SU-processen. Man definierar och avgränsar ett antal delområden inom systemutveckling och gör därmed en *fasindelning*. Det finns många olika SU-modeller. Man kan t ex nämna den klassiska SIS/RAS-modellen (SIS, 1973) som det har gjorts många företagsanpassningar av. Det finns nu en ny referensmodell för systemutveckling (SIS, 1989). Detta är exempel på två SU-modeller, som alltså innehåller en *strukturering* och *sekvensiering* av systemutveckling i ett antal delområden (faser).

Dessa modeller avser att tala om vad som ska göras i SU-arbete, men inte hur det ska göras; dvs modellerna ska vara metodoberoende. Modellen ska ange en struktur av områden. I ett område ska det gå att stoppa in olika metoder. En kritisk fråga är dock om det överhuvudtaget går att etablera en övergripande SU-struktur som är helt metodoberoende. Bygger inte varje modellstruktur på någon underförstådd metodsyn? Detta var en kritik som bl a riktades mot den klassiska SIS/RAS-modellen och samma kritik anser jag man kan rikta mot den nya referensmodellen.

Att det finns en underförstådd metodsyn bakom en modellstruktur innebär dock inte att modellen skulle vara helt metodberoende. Man kan istället tala om *grad av metodberoende*, och där vissa modeller säkert kan sägas ha en hög grad av metodberoende. En viss klass av metoder kanske är mer förenliga med vissa modeller än andra.

En exemplifiering för förtydligande: Det finns modeller som uttryckligen ansluter sig till en datadriven ansats (t ex SVEA; se Axelsson-Ortman, 1985) - sk information engineering-syn - och detta sätter givetvis sina spår i modellstrukturen: En övergripande datamodelleringsfas bör ingå på tidigt stadium i modellen.

Detta visar alltså modellens beroende av vissa grundläggande metodantaganden. Modellen bygger på ett bakomliggande synsätt.

En annan form av kritik som riktades mot SU-modell av typ SIS/RAS var just deras metodoberoende eller snarare brist på metodmässigt innehåll. Praktiska företagsanpassningar av t ex SIS/RAS innebar att modellen fylldes med mer metodmässigt innehåll. Så småningom började också mer kompletta modeller med ingående metoddelar att marknadsföras. (Detta kommer att behandlas vidare nedan i avs 2.3).

Som en konsekvens av detta har termen SU-modell blivit tvetydig. Ibland används "modell" för en (i princip) metodoberoende modellstruktur och ibland för en modellstruktur fylld med specifikt metodinnehåll. För det senare har ibland termen *metodkedja* använts. Ofta innebär dock *metodkedja* en sammanhängande integration av olika metoddelar (Fåhraeus, 1986).

När man fyllt en övergripande modell med metodmässigt innehåll och därmed kanske bildat en metodkedja så erhåller man ofta en underliggande, mer detaljerad fasstruktur. Varje "huvudfas" kommer att bestå av ett antal delfaser. Modellstruktur är alltså något man kan se hierarkiskt. En övergripande struktur och indelning i faser/områden som är relativt metodoberoende; och en uppdelning av varje område i mindre delfaser/delområden. Denna uppdelning på lägre detaljeringsnivå innebär förmodligen i de flesta fall ett ökat metodoberoende.

En fas eller område innebär att man anger en arbetsuppgift eller klass av arbetsuppgifter.

Istället för den nu tvetydiga termen modell föredrar jag termen *modellstruktur* (för systemutveckling) för att benämna

"uppdelning i och ordning av delområden/faser som systemutveckling eller del därav består av."

När jag nedan talar om *SU-modell* avser jag modellstruktur med eller utan metodmässigt innehåll.

1.22 Metod

Genom modellstruktur talar man om vad som ska göras, men inte hur det ska göras. Metoder är vägledningar för det praktiska arbetet med systemutveckling. Metoder innehåller riktlinjer för analys/utformning och beskrivning. Detta innebär föreskrifter för hur arbetet ska utföras; t ex vilka typer av frågor som ska ställas, vilka faktorer som ska beaktas, vilka typer av analys/konstruktionsresonemang som ska tillämpas, vilka kriterier som

3

ska användas. En metod innehåller alltså regler för arbetssätt.

Systemutveckling kan beskrivas som ett problemlösande och kunskapsutvecklande design-arbete. Man kan säga att en metod talar om vilka typer av frågor som bör ställas i SU-arbetet och hur man ska uttrycka och beskriva svaren. SU-metoder innehåller vanligtvis regler för olika typer av beskrivningar. Ofta är arbetssätt och beskrivningssätt väl integrerat i metoderna. Beskrivning av ett arbetsmoment inom en metod (ofta kallat metodsteg) inrymmer ofta föreskrifter för både arbetssättet och hur man ska beskriva resultatet av arbetssteget. Ibland kallas delar av SU-arbete för modellering; just med syfte att framhäva att resultatet av arbetet ska bli en modell, en beskrivning.

Jag vill dock här undvika att använda ordet *modell* för de beskrivningar av system/verksamhet/utrustning mm som framställs under SU-processen. Jag gör det för att undvika sammanblandning med användning av termen *modell* i systemutvecklingsmodell och modellstruktur; se ovan avsn 1.21. Istället använder jag huvudsakligen orden *beskrivning*, *dokument* eller *dokumentation* för sådana resultat av SU-arbete.

Riktlinjer för beskrivning brukar kallas för beskrivningsteknik, modelleringsteknik, dokumentationsform eller notation. Ofta avses beskrivningstekniska regler för olika grafer; t ex för ISACs V-grafer (Lundeberg m fl, 1978) eller MBI-metodens verksamhetsgrafer (Hugoson m fl, 1983). Notation är dock inte begränsat till grafiska framställningssätt. Det kan även gälla tabeller, förteckningar, matriser eller andra beskrivningssätt.

Genom notationen definierar man aktuell beskrivningsform. Man talar om vilka slags fenomen (objekt) som ska beskrivas samt vilken framställningsmässig form de ska ha. Man talar också om hur olika beskrivningselement får kombineras. Genom notation anger man för en viss typ av beskrivning dess

- semantik (vilka beskrivningselement)
- uttrycksform (hos beskrivningselement)
- syntax (regler för kombination av beskrivningselement)

Notation innehåller regler för *framställning* och *tolkning* av beskrivningar.

Ett exempel för förtydligande: Notation för ISACs V-grafer består av angivande av vad som beskrivs i sådana grafer; "materiella mängder, meddelandemängder, flöde av mängder och delverksamheter". Notationen talar också om uttrycksform; "meddelandemängder beskrivs grafiskt med romboider". Det finns också regler för hur beskrivningselement får kombineras; t ex "delverksamheter får inte sammanbindas utan mellanliggande matriell mängd eller meddelandemängd".

En viktig del i notationen är alltså de typer av beskrivningselement som används. De innebär definition av de företeelser som man ska inrikta sitt analys/konstruktionsarbete på under SU. Härigenom ger metoden en stark styrning av vad som uppmärksammas och talas om under SU-arbetet. Metoden ger en stark *normativ konceptualisering* för systemutvecklingen. Med konceptualisering menas de begrepp och kategorier (t ex

materiellt flöde, dataflöde, dialogstruktur, normaliserad relation) som inblandade aktörer använder när de kommunicerar med stöd av viss metod. Konceptualisering innebär hur man identifierar, avgränsar, uppfattar och benämner relevanta kategorier i den verklighet som förändras och utformas. Metoden och särskilt dess notation har en stark påverkan på hur man konceptualiserar (begreppsliggör) aktuell verklighet. Metodens använda grundbegrepp har alltså en starkt styrande inverkan på människors tänkande, och därför finns en tydlig normativ funktion.

Begrepp som används i notationen finns ofta med i beskrivningar av föreslaget arbetssätt; dvs vilka frågor som ställs. Ett exempel: Begreppet *problem* är centralt i metoden förändringsanalys/SIM (Goldkuhl & Röstlinger, 1988). Det finns med i metodbeskrivningen av arbetssätt; olika arbetsmoment som "identifiering och formulering av problem, analys av problemsamband". Det finns också med som centralt beskrivningselement i notationen; i "problemlista, problemgrafer".

Mot bakgrund av detta kan man säga att metod består av tre (ofta väl) integrerade delar:

- arbetssätt
- begrepp
- notation

Metodes funktion är att vägleda SU-arbete. Ofta är olika "metoddelar" sammankopplade till en mer omfattande metod. Några exempel på detta är ISAC, REFLEX, SVEA, LOGIC och SIM. För att tala om en sådan sammanhängande mer omfattande metod används ofta termer som "metodkedja" eller "metodik".

Ett annat ord i sammanhanget är *metodologi*. Med metodologi vill jag avse "läran om metoder", dvs ingen specifik metod utan istället samlad kunskap om metoder. Det är närmast det jag nu som författare ägnar mig åt - "metodologisering" - studium av metoder som företeelse. Se även t ex Nilsson (1991) och Andersen (1991) för diskussion om olika metodbegrepp.

1.23 Verktyg

Under de senaste åren har det skapats många olika typer av datorverktyg (programvaror) som stöd åt SU-processen. Först infördes datorstöd för sena faser (programmering/realisering). Datorstöd har nu även nått de tidiga faserna av systemutvecklingsprocessen; behovsanalys etc.

En viktig typ av datorstöd i detta sammanhang är sk 4e generationens språk (4GL) eller ibland kallade informationshanteringssystem (IHS-system); se Lie-Nielsen & Colliander (1983) och Söderström & Olstedt (1988). 4GL används bl a för att snabba upp och "rationalisera" utvecklingsarbetet. 4GL-verktyg används också för att på tidigt stadium illustrera aktuellt system eller delar därav; sk prototyping.

Andra viktiga verktyg som framkommit under de senaste åren är sådana som stödjer analys- och konstruktionsprocessen under systemutveckling. Sådana hjälpmedel kallas ofta för CASE-verktyg (Söderström mfl, 1990). CASE är en förkortning för "Computer-Aided Systems Engineering" (eller ibland "Software" istället för "Systems").

Ofta innebär CASE-verktyg stöd åt en eller flera specifika metoder. Verktöget stödjer framtagande av beskrivningar (t ex grafer) enligt metoden (dess notation). Man strukturerar SU-informationen i enlighet med metodens begrepp. Att arbeta med CASE-verktyg innebär således datorstödd metodbaserad systemutveckling.

CASE-verktyg består ofta av

- grafiska editorer (för framställning/hantering av grafiska beskrivningar)
- designdatabas ("dictionary, repository"; för central informationshantering av designobjekt/begrepp)
- funktioner för olika analyser och kontroller av dokumentationen
- transformationsfunktioner (mellan olika designbeskrivningar eller generering av databas-/programkod eller generering från kod, sk "reverse engineering")
- sök- och rapporteringsfunktioner (färdiga eller genererbara)
- funktioner för import från/export till andra verktyg
- projektadministrativa stödfunktioner

Se vidare t ex Söderström m fl (1990) eller Hällström (1989).

1.24 Synsätt

Alla modeller och metoder bygger på något bakomliggande synsätt. Detta synsätt behöver inte vara särskilt tydligt uttryckt i samband med modellen eller metoden. Det kan vara underförstått hos de personer som utvecklar, marknadsför eller använder modellen/metoden.

Ofta är vi inte helt medvetna om de antaganden och uppfattningar som vi har om aktuell verklighet (Lindholm, 1979; Goldkuhl, 1986). Vi tar för givet det sätt som vi ser på verkligheten och att det är sättet att se på verkligheten. Vi är inte alltid medvetna om att det finns olika synsätt som kan anläggas på studium av verkligheten.

Det synsätt som en modellkonstruktör har på SU-verkligheten kan vara delvis

- omedvetet
 - oreflekterat
 - otydligt
 - otillräckligt artikulerat
- samt därmed innehålla
- förutfattade föreställningar.

6

Synsätt är det som ligger bakom och styr en modell eller metods innehåll och utformning. Synsättet utgör motivering till metoden. Synsätt kan sägas innehålla

- principer,
- värderingar,
- föreställningar,
- erfarenheter och
- kategoriseringar
- definitioner

som är relevanta för SU-arbete.

Ett mer akademiskt sätt att uttrycka vad som menas med synsätt är att säga att det är teori eller perspektiv på systemutveckling (Nurminen, 1988); och där denna teori är både normativ, förklarande och klassificerande till sin karaktär.

För att förtydliga vad som menas med synsätt redovisar jag några "etiketter" som används för att ange olika aktuella synsätt inom området:

- affärsinriktad
- verksamhetsinriktad
- datadriven
- funktionsdriven
- händelsedriven
- objektorienterad
- teknikdriven
- problemdriven
- förändringsprocessinriktad
- användarorienterad
- ingenjörsbaserad
- analytisk
- experimentell

Man kan se dessa som olika karaktäriseringar på systemutveckling och som har sitt ursprung i olika sätt att betrakta SU-processen på. Detta innebär att det är benämningar på delvis olika synsätt på systemutveckling.

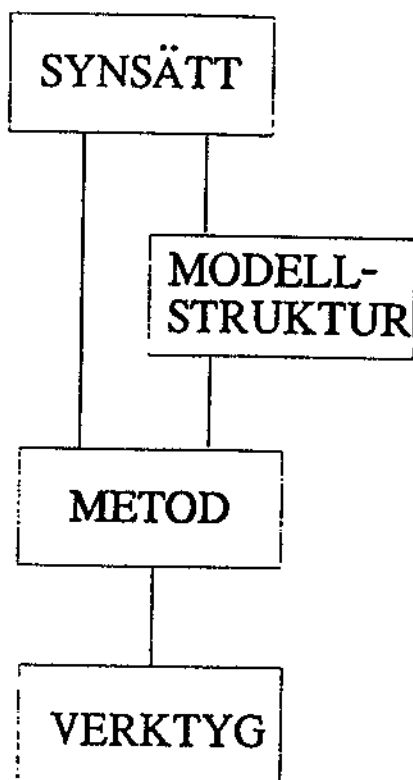
1.3 EN BEGREPPSSTRUKTUR

Dessa olika företeelser

- synsätt
- modellstruktur
- metod
- verktyg

kan ses som hjälpmedel för att ge bättre stöd och struktur åt SU-processen. De är hjälpmedel för bättre förståelse och handling. Dessa olika hjälpmedel har relationer till varandra. De kan sägas samspela i praktiskt SU-arbete. Hur ska man se på relationerna mellan dem?

Jag presenterar nedan (fig 1.1) en strukturell beskrivning.



Figur 1.1 Systemutveckling: Strukturbeskrivning av olika stödjande företeelser

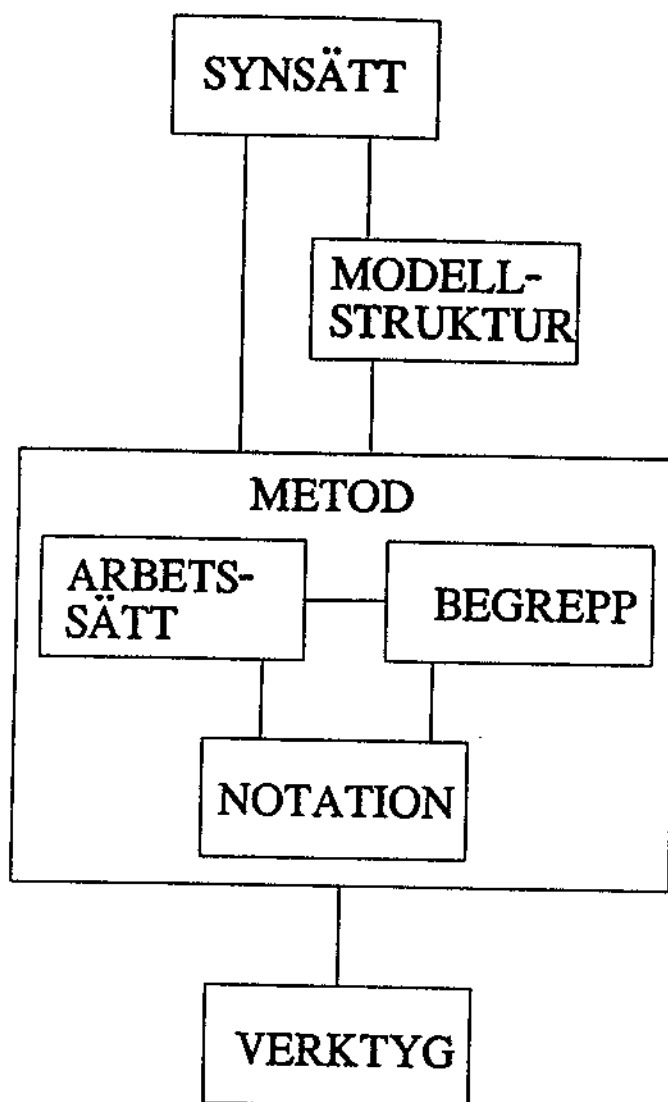
Synsätt är något som ligger bakom modellstrukturering och metod. Indelning av SU i områden/faser bygger på något sätt att konceptualisera SU på. En metod bygger helt klart på ett eller flera SU-synsätt. Man kan se metoden som ett hjälpmedel att förverkliga ett specifikt synsätt.

Ett exempel: Jag har själv presenterat ett synsätt "verksamhetsinriktad systemutveckling" och försökt redovisa på vilka sätt detta synsätt kan realiseras i praktiskt SU-arbete med hjälp av metoden verksamhets- & informationsbehovsanalys (VIBA) enligt SIM (Goldkuhl, 1989,1990b).

En metod kan användas i en viss fas (område) i en modell- (struktur). En metod kan på detta sätt ingå i en eller flera modellstrukturer.

CASE-verktyg är som beskrivits ovan (avs 1.23) hjälpmedel för att utföra SU enligt viss metod eller uppsättning av metoder.

I figur 1.1 beskrevs metod som ett samlat begrepp. En mer tydlig beskrivning ges i figur 1.2 där dess ingående delar dessutom anges. Synsätt har som nämnts en stor betydelse för metod. Speciellt vissa av de begrepp som ingår i metoden har säkert sitt "ursprung" i ett visst synsätt.



Figur 1.2 Systemutveckling: Fördjupad strukturbeskrivning av olika stödjande företeelser

Ett exempel: förändringsanalys enligt SIM-metoden (Goldkuhl & Röstlinger, 1988) bygger på ett artikulerat problemlösnings- och förändringsperspektiv. I detta synsätt ingår t ex *problem* och *förändringsåtgärd* som väsentliga begrepp. Dessa begrepp ingår tydligt i metoden. Man kan säga att metoden innehåller vissa begrepp som direkta konsekvenser av ett visst synsätt. På detta sätt påverkas både arbetssätt och notation. Som nämndes tidigare (avs 1.22) innehåller både arbetssätt och notation t ex begreppet *problem*.

Synsätt påverkar givetvis också arbetssätt på ett direkt sätt. Olika värderingar och principer i synsättet kommer att ha konsekvenser i metodmässiga riktlinjer för arbetssätt.

Ett exempel: SVEA-metoden (Axelsson-Ortman 1985) och dess vidareutveckling IRMA & SVEA (Swende 1989) bygger på en data-

driven syn. Den data-drivna synen (sk IRM-syn) inbegriper bl a starka principer för dataoberoende och dataintegration (se t ex Griethuysen, 1982). Dessa principer har kommit klart till uttryck i själva arbetssättet för IRMA & SVEA; bl a genom en verksamhetsövergripande datamodellering som sker på tidigt stadium och som styr mycket övrigt efterföljande SU-arbete.

Synsättet kan också direkt påverka notationen.

Ett exempel: Som grund för ISAC-metoden (Lundeberg m fl, 1978) fanns ett uttalat intresse att stödja användarinflytande. Detta har påverkat beskrivningsteknikernas utseende och innehåll. Man eftersträvade relativt enkla grafiska beskrivningsformer. Dessa är renodlat "informations-/verksamhetsorienterade" och därmed utan några datatekniska inslag som kräver sådan kompetens.

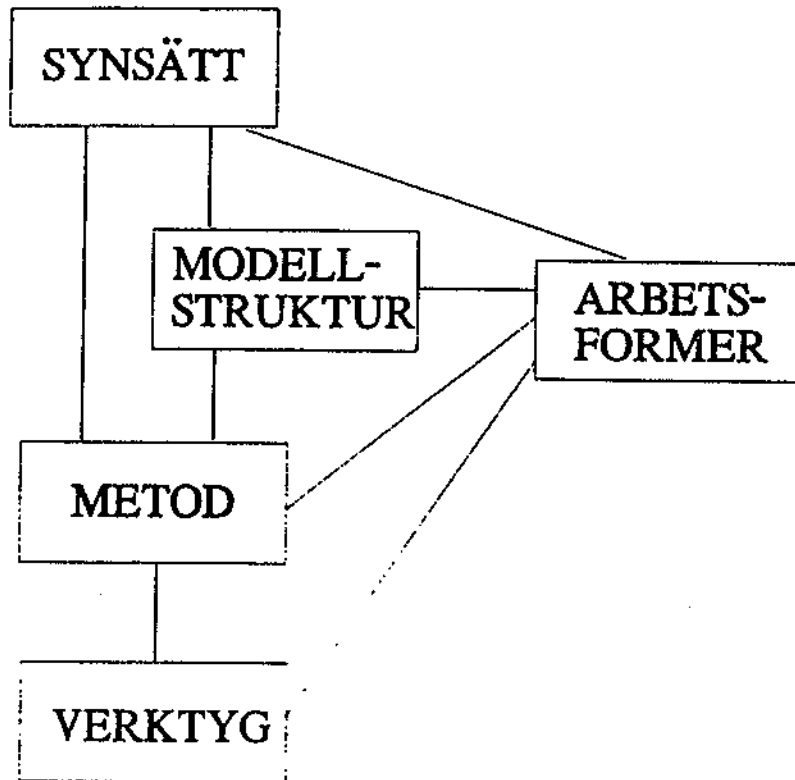
Den hittillsvarande beskrivningen har haft ett fokus på metodfrågor. Detta är inte hela bilden av systemutveckling. Det finns andra viktiga dimensioner också. En viktig aspekt som jag inte kan undvika att nämna här, för att göra bilden mer rikhaltig, är arbetsformer under systemutvecklingen.

Arbetsformer har stor betydelse för framgång i systemutveckling. Detta berör bl a hur roll- och arbetsfördelning är mellan olika parter i SU-processen. Man talar ibland om olika förändringsstrategier för att betona att rollfördelning kan vara olika (se t ex Targama, 1981):

- expertstrategi
- förankringstrategi
- processtrategi

Det finns många författare som hävdar betydelsen av att SU-arbetet sker med ordentligt användarinflytande; se t ex Mumford & Henshall (1983), Andersen m fl (1986) och Utvecklingsprogrammet (1988). Samtidigt finns det många problem förknippade med användarinflytande som bl a Nurminen (1988) identifierat och beskrivit. Användarinflytande kan inte kompensera bristfälligt arbetssätt eller synsätt!

Det är viktigt att de metoder som man arbetar med harmonierar väl med de arbetsformer som tillämpas (Goldkuhl & Röstlinger, 1988; Selldén, 1988). I figur 1.3 har jag kompletterat den tidigare strukturbeskrivningen med arbetsformer.



Figur 1.3 Systemutveckling: Kompletterad strukturbeskrivning av olika stödjande företeelser

2 NÅGRA HISTORISKA UTVECKLINGSTENDENSER

Med ovanstående referensram som grund ska jag i detta kapitel ge några historiska utvecklingstendenser. I nästa kapitel ska jag titta lite framåt. Denna historiebeteckning blir - bl a av utrymmesskäl - ganska fragmentarisk. Jag vill lyfta fram några saker som jag anser är intressanta som underlag för den efterföljande framåtblickande analysen.

2.1 OLIKA UTVECKLINGSMETODER

Sedan i början av 1970-talet har olika metoder för systemutveckling successivt utvecklats och börjat tillämpas. Idag tar många människor för givet att metoder ska användas vid utveckling av informationssystem. Annat var det på tidigt 70-tal! Jag har själv varit aktiv i metodarbete sedan 1973 och minns väl hur skeptiska många erfarna systemerare var inför förekomsten av olika systemeringsmetoder. Systemerare hade genom mångårig erfarenhet baserad på "trial and error" byggt upp en egen personlig arbetspraxis som åtminstone i egna ögon ansågs fungera. Mot denna bakgrund var man ofta skeptisk mot metoder som upplevdes som "nymodigheter" av begränsat värde.

Emellertid har ett mer metodbaserat arbete fått ökad genomslag i praktiken. Olika utvecklingsmetoder har successivt utvecklats och marknadsförts. Jag har redan i avsnitt 1.22 nämnt olika sådana metoder. Ett exempel är ISAC-metoden (Lundeberg m fl, 1978) som det bedrevs omfattande metodutveckling och metodprövning för under 70-talet. MBI-metoden (Hugoson m fl, 1983) utvecklades också under 70-talet och blev så småningom integrerad i MBI/SAK. Fortsatt vidareutveckling har skett vilket lett fram till REFLEX.

Under 80-talet skedde utveckling av SVEA-metoden (Axelsson-Ortman, 1985). Den kompletterades så småningom till IRMA & SVEA (Swende, 1989). Denna metod finns även i en något modifierad form som kallas DIRECT (Axelsson-Ortman, 1990).

Ingen omfattande vidareutveckling av ISAC har skett under 80-talet. ISAC-metoden har kompletterats med en metod för val och anpassning av standardsystem som kallas SIV (Nilsson, 1991).

Jag var själv delaktig i utveckling av ISAC-metoden under 70-talet, men har därefter tagit andra vägar vad gäller SU-metodik. Under 80-talet har jag aktivt arbetat med utveckling med SIM-metoden för förändringsarbete och verksamhetsinriktad systemutveckling (Goldkuhl & Röstlinger, 1988; Goldkuhl, 1989, 1990ab). Denna metodutveckling har bl a baserats på en kritisk analys av erfarenheter med ISAC-metoden (Goldkuhl, 1984).

Ovanstående har mest varit uppräknings av olika namn på metoder. Men vad innehöll de då? Här är givetvis inte platsen för någon omfattande genomgång av metodernas innehåll; istället hänvisas till ovan refererad metodlitteratur eller översiktsverk som t ex Andersen (1991). Några metod ingredienser ska dock nämnas.

Jag vill först påpeka att jag i min genomgång av SU-metoder har gjort en avgränsning till främst metoder för "analys-faser" av systemutvecklingen. Under denna tid har också flera metoder för tekniskt design-arbete sett dagens ljus; t ex för databasdesign och programkonstruktion. För det senare finns JSP-metoden som fått en mycket stark ställning på svenska marknaden. JSP har senare vidgats till en metod för SU, JSD (Ingevaldsson, 1985), och även denna har introducerats på den svenska marknaden. JSD är en metod för analys av "objekt och händelser".

Bland tidiga metoder fanns en fokus på metoder för *informationsanalys* och *verksamhetsanalys*. Dessa var i allmänhet grundade på systemteori (Langefors, 1966) med hierarkiska arbets- och beskrivningssätt som följd. Exempel på detta är ISACs V-grafer och I-grafer samt MBI.

Så småningom ökade kunskapen och intresset för *informationsmodellering/datamodellering*. Den sk SASMO-metoden (Swende, 1982) utvecklades bl a på basis av Sundgrens (1981) arbeten och andra internationella arbeten kring databasdesign, och då främst med utgångspunkt i relationsteori. Sk normalisering kom att vara ett väsentligt inslag i detta.

SASMO-inspirerad datamodellering införlivades sedan i SVEA-metoden. Det har sedan införlivats i flera andra metoder och har nog kommit att betraktats som metodmässigt allmängods nuförtiden. Det existerar dock flera olika varianter på metoder för informationsmodellering (Dahlgren m fl, 1991). Det har också kommit att tillämpas med väsentligt olika förtecken i olika metoder. Ibland handlar det om att göra en applikations-övergripande datamodell för gemensam styrning av företagets informationsresurser - en "stadsplan" (Swende, 1989). I andra fall görs modellering av enstaka lokala system eller t o m begränsade delar därav. För kritisk analys av datamodellering hänvisas till Lyytinen (1983), Kall (1984) och Magoulas (1991).

En annan utvecklingslinje var ett begynnande intresse för ärendebeskrivningar till skillnad från de funktionellt och hierarkiskt inriktade metoderna för verksamhetsanalys (som nämndes ovan). Man ansåg att metoderna för verksamhetsanalys inte klarade av att beskriva olika detaljerade förlopp i verksamheten; framför allt interaktionen mellan informationssystemet och dess användare. Olika ansatser för sk *rutinskissning* har presenterats. Rutinskissning ingår i SVEA/DIRECT men också i andra metoder. Den norska Vagg-grafmetoden (finns bl a beskriven av Andersen, 1991) var ett tidigt exempel på ärendebeskrivande metod och har förmodligen inspirerat andra senare metoder.

Nilsson (1991) menar att detta är två skilda dimensioner i utvecklingsarbetet - funktionsanalys och händelseanalys - och som därmed också behöver hållas åtskilda. Handlingsgrafer (Goldkuhl, 1990a) som ingår i SIM-metoden innebär en ansats som försöker förena och därmed överskrida separationen mellan funktionsanalys och ärendeanalys. De funktionellt orienterade metoderna för verksamhetsanalys är hierarkiskt uppbyggda och

bygger som sagt på några av systemteorins grundläggande principer. Detta är dock inte det enda sättet att bedriva verksamhetsanalys på. Handlingsgrafer i SIM-metoden innebär istället ett sk kontextuellt angreppssätt. Detta innebär kortfattat att man som beskrivare själv väljer lämplig avgränsning av dokument och låter olika dokument "haka" i varandra genom enkla konnektorer istället för att vara beroende av hierarisk beskrivningsstruktur. Man tillämpar "bottom all" istället för "top down".

En annan utvecklingslinje var att söka sig tidigare i utvecklingskedjan. I ISAC identifierades och myntades konceptet förändringsanalys. Förändringsanalys innebär en undersökning av om utveckling av informationssystem är att betrakta som lämplig förändringsåtgärd i den aktuella situationen, dvs ett diagnosarbete. I princip kan man säga att förändringsanalys föregår beslut om och genomförande av systemutveckling.

ISAC-metoden för förändringsanalys (Lundeberg m fl, 1978) kan dock anses som ofullständig (Goldkuhl, 1984). SIM-metoden för förändringsanalys utvecklades senare och innehåller bl a viktiga moment av problemanalys och målanalys. Dessa har också senare börjat tas upp i andra metoder.

Jag har ovan kortfattat skisserat en utveckling med utgångspunkt tagen i svenska förhållanden. Det har givetvis skett ett omfattande metodarbete på andra håll i världen. Jag har bl a nämnt datamodellering som är en stark internationell trend och där starka influenser utifrån märks. Datadriven ansats med datamodellering sätts ibland som motsats till sk funktionsdriven ansats. Internationellt spelar här sk funktionsmodellering med dataflödesdiagram en stor roll, ofta kallat Strukturerad Analys (Yourdon, 1989; Gane & Sarson, 1979). På senare år har dessa metoder börjat få ett fotfäste på den svenska marknaden.

Dessa metoder kan jämföras med svenska metoder för verksamhetsanalys (ISAC, MBI) som också är funktionellt och hierarkiskt uppbyggda. De svenska metoderna är dock bredare till sin karaktär då de även inbegriper analys och beskrivning av materiella flöden vid sidan om informationsflöden.

2.2 ÖKAD ANVÄNDNING AV VERKTYG

I avsnitt 1.23 ovan beskrevs introduktion av olika verktyg för systemutveckling. I slutet av 70-talet ökade intresset för prototyping starkt. Man talade om systemskisser, experimentell systemutveckling och pilotimplementering som ansatser inom området (se t ex Persson, 1977). Man använde ofta högnivåspråk som delvis kan ses som embryon till senare 4e generationens språk (4GL). Under 80-talet skedde sedan utveckling och marknadsföring av mer renodlade 4GL-produkter som innebär en integrering av flera olika funktioner till en total utvecklingsmiljö. I ett 4GL ingår ofta förutom ett kraftfullt högnivå-programspråk även relationsdatabashanterare, frågespråk, rapportgenerator, formulärhanterare och liknande

funktioner. 4GL har kommit att användas både för prototyping (utvecklingsmiljö) och för slutlig realisering (målmiljö).

4GL kan sägas ha varit en "modetrend" under åtminstone första delen av 80-talet. Under senare delen har konkurrens kommit från en annan teknik-trend, nämligen CASE-verktyg. CASE-verktyg innehåller ibland 4GL-funktioner eller har åtminstone kopplingar till sådana andra verktyg.

CASE-verktyg innebär som nämnts ovan att en eller flera metoder blir datormässigt implementerade. Detta gäller åtminstone verktyg som stödjer tidiga faser (sk "Upper-CASE"). Vilka metoder är det då som har implementerats i de många CASE-verktyg som nu marknadsförs?

Slående är hur lika de flesta verktygen är. De stödjer vanligtvis några amerikanska standardmetoder från 70-talet. Det är främst funktionsstrukturering med dataflödesdiagram som dominerande notation samt olika notationer för datamodellering. Detta är i sig inte särskilt märkligt eftersom många CASE-verktyg har ett amerikanskt ursprung.

Hur stämmer detta med den svenska (och skandinaviska) metod-traditionen? För att på ett enkelt sätt karaktärisera den svenska traditionen, med utgångspunkt i avsnitt 1.22 och 2.1 ovan, så kan man säga att den är bredare och mer verksamhetsorienterad i relation till amerikansk tradition. Man har t ex här en starkare orientering mot verksamhetsanalys och även ofta intresse för olika arbetsorganisatoriska konsekvenser. Jag vill karaktärisera amerikanska metoder som snävare och mer tekniska till sin inriktning.

Därför finns det givetvis en fara med den kraftiga marknadsföring av ofta amerikanska CASE-verktyg (innehållande amerikanska standardmetoder) som nu sker. Vad händer med det mer verksamhetsinriktade tänkandet som kännetecknar svensk SU-tradition när man börjar arbeta med stöd från dessa amerikanska CASE-verktyg?

Om vi återknyter till resonemanget från avs 1.3 ovan (om begreppsstrukturen): Verktyg ska stödja metoden, dess arbets-sätt och notationer. Verktyget är ett hjälpmedel i relation till metod. Detta innebär att man bör välja verktyg som passar till det önskade metodmässiga arbetssättet och inte tvärtom. Så är dock inte alltid fallet. Jag har tagit del av flera berättelser där man valt verktyg och sedan som konsekvens fått ändra metod. Det finns en viss tendens att utvecklingen inom SU-området styrs "underifrån", från tillgänglig teknik, istället för vad man vill uppnå, dvs ur syftes- och kvalitets-synvinkel. Kort sagt, att medlen bestämmer målen!

Kan man då inte utveckla CASE-verktyg för svenska metoder? Jovisst, och så har också skett. Relativt tidiga satsningar var DASAK för SAK-metoden och GraphDoc för ISACs V-grafer (Boström mfl, 1986). Detta var innan begreppet CASE överhuvudtaget kommit till Sverige. Dessa satsningar innebar att verktygen programmerades från grunden. Senare satsningar har tagit ett annat grepp. Man har där utnyttjat sk CASE-skal, vilket är en

mycket viktig företeelse inom området; se Bubenko (1989), Lyytinen m fl (1989), Goldkuhl (1991).

CASE-skal (som också kan kallas meta-CASE-verktyg eller CASE-generator) är ett verktyg med vars hjälp man utvecklar ett CASE-verktyg som stödjer metoder som man önskar använda. CASE-skal är alltså ett slags design-verktyg med vars hjälp man med "överkomlig" resursinsats kan tillverka sitt eget CASE-verktyg, utan att behöva bygga verktyget helt från grunden i något sedvanligt programspråk.

Baserat på CASE-skal har det har utförts flera metodanpassningar av CASE-verktyg för svenska metoder. Det är främst två CASE-skal som har utnyttjats: Det amerikanska Excelerator/Customizer och det svenska Ramatic, som har utvecklats av forskningsinstitutet SISU (Bergsten m fl, 1989).

Några exempel: Excelerator/Customizer har bl a utnyttjats för att ta fram ett verktyg för REFLEX (kallas REX) och ett för DIRECT (kallas DAX). Ramatic har bl a utnyttjats för att ta fram ett verktyg för SVEA (kallas SVAR). Det har även använts för att ta fram ett verktyg för en företagsanpassad version av LOGIC (på Volvo PV).

Det finns även andra intressanta CASE-skal som kan utnyttjas. Nämnas kan VSF, Design/OA och MetaEdit.

Metodanpassning av CASE-verktyg baserat på CASE-skal är ett relativt nytt och mycket intressant kunskapsområde. I forskningsprojektet PRIMCASE studerar vi detta område (Goldkuhl, 1991). PRIMCASE ingår i Fou-gruppen VITS (Goldkuhl & Röstlinger, 1991).

2.3 KONSULTMODELLER SOM KOMBINERAR OLIKA METODER

Jag beskrev i avs 2.1 ovan framväxten av olika metoder i Sverige. Det är inte bara metoder som har utvecklats utan också SU-modeller. SIS/RAS satte en stor prägel på detta under 70-talet. Det gjordes många företagsanpassningar av SIS/RAS. I samband med detta fylldes det metodlösa ramverket med metodinnehåll. Jag uppfattar att med tiden så har användningen av SIS/RAS som modell mjukats upp och sedan ofta övergivits.

Det har kommit många konkurrenter till SIS/RAS, som förutom modellstruktur ofta innehåller ett antal metoder. Många av de ovan (avs 1.22 och 2.1) nämnda metoderna innebär i själva verket en modell med vidhängande metoder. Detta gäller REFLEX, SVEA, DIRECT och LOGIC. Dessa modeller har utvecklats och marknadsförs av olika konsultföretag. Det finns ytterligare sådana modeller.

Det som i allmänhet är specifikt för dessa modeller är den övergripande modellstrukturen. Det metodmässiga innehållet är ofta hämtat från annat håll; t ex datamodellering. LOGIC-modellen t ex, uppfattar jag som ett "hopkok" av olika metoder med ibland lättare modifieringar. LOGIC innehåller bl a datamodellering, dataflödesdiagram enligt Yourdon, problemanalys

från FA/SIM. För att fortsätta med maträtts-metaforen så innebär LOGICs hopkok att ingredienserna (metoderna) är hämtade utifrån medan receptet (modellstrukturen) har man gjort själv. Jag vill betona att LOGIC här har använts som exempel, men samma sak gäller i princip för de andra nämnda modellerna i större eller mindre grad.

3 NÅGRA FRAMTIDA UTVECKLINGSTENDENSER

Efter att ha blickat bakåt - och försökt göra ett begripligt mönster av det gytter av historiska skeenden - är det nu dags att blicka framåt. Det är givetvis ännu svårare. Vilka är de kommande utvecklingslinjerna? Det blir naturligtvis spekulationer. Något annat är inte möjligt. Men jag försöker grunda mina hypoteser om framtiden genom att följa och dra ut de historiska utvecklingslinjer som jag ovan har beskrivit. Jag skisserar några tänkbara utvecklingslinjer.

Min framtidsbild är förstas även den fragmentarisk. Jag tar nedan upp några intressanta utvecklingstendenser som handlar om synsätt, modellstruktur, metod och verktyg samt samspelet mellan dessa.

3.1 SYNSÄTTENS ÖKANDE BETYDELSE

Inom forskningsvärlden har sedan länge förts diskussioner om olika synsätt, deras fördelar och nackdelar; se t ex Nurminen (1988), Hirschheim & Klein (1989), Magoulas & Pessi (1991), Dahlbom & Mathiassen (1991). Jag tror att denna diskussion kommer föras allt mer i praktikfältet också. Introduktion av nya synsätt - t ex objektorientering - tvingar fram detta.

Med detta följer en ökad medvetenhet om synsättens betydelse och att man kanske inte kan blanda dem för mycket som nu görs i olika modeller. Man blandar i vissa modeller metoder som bygger på väsentligt skilda synsätt, t ex funktionsdrivet och data-drivet. Risken finns att man får en dålig kompromiss. Man kommer kanske inse att man måste välja något synsätt som överordnad princip.

När jag talar om synsättens ökande betydelse menar jag också att man klarare ska se metodernas grunder; vad de bygger på för antaganden, värderingar och principer.

Många marknadsförda modeller presenteras ofta i handboksform (i "metodpärmar"). Detta gäller också olika företagsinterna modell/metodversioner. När jag studerat olika sådana handböcker tycker jag det är slående hur starkt orienterade de är mot "hur man gör". Det finns i allmänhet väldigt lite om "varför man ska göra så och så", dvs motiv och bakomliggande synsätt. Metodhandböcker är tyvärr alltför ofta bara "bruksanvisningar" utan tillräcklig genomlysande argumentation varför man ska utföra ett visst moment, och just på det föreskrivna sättet. Det finns då en stor risk att metodanvändningen blir mekanisk; man följer metodens bokstav utan att djupare förstå metodens anda och synsätt. När man då stöter på svårigheter i SU-arbete finns risk att man alltför lättvindigt slutar använda metoden då man inte klarar av att situationsanpassa metodanvändningen. En mekanisk metodanvändning kan också säkert leda till att man inte uppnår de effekter (kvalitetskrav) som tillämping av metoden bör leda till enligt synsättet.

Jag argumenterar här för önskemålet om mer förståelseorientering i metodbeskrivningar, men även i samband med metodutbildning.

3.2 PENDLING MELLAN STANDARDISERING OCH INNOVATION

Införande av modeller, metoder och verktyg har bl a syfte att standardisera SU-arbete. Man vill uppnå mer enhetliga arbetsätt. Genom enhetliga arbetssätt så minskas personberoendet. Man minskar sitt beroende av systemerares speciella arbets- och dokumentationsstil. Genom att man använder samma metoder så kan en person enklare ersättas. Enhetliga arbets- och dokumentationssätt spelar roll för det direkta SU-arbetet. Genom att inblandade personer delar samma metodkunnskap kan kommunikation, samarbete och gemensam problemlösning befrämjas.

Man kan också öka lärandet mellan olika projekt och personer. Den gemensamma referensramen underlättar bearbetning och utbyte av erfarenheter. Man talar samma metodspråk. Det finns en stor potential i ett gemensamt metodspråk i en organisation. Detta har lett till att personer ibland säger: "Det spelar ingen roll vilken metod man har, bara man har en metod". Men metoder har olika egenskaper. Det finns bättre metoder och sämre metoder. Varför ska man välja en sämre metod framför en bättre?

En metod kräver likformighet på gott och ont. Som systemerare kan det ibland innebära inskränkningar i möjlighet till individuellt anpassat arbetssätt och därför försämra motivationen. En ökad användning av CASE-verktyg minskar givetvis utrymmet för personliga metodvarianter. Detta kan innebära att det "konstnärliga" i systemeraren får stå tillbaka på bekostnad av det "ingenjörsmässiga" (Malmborg, 1991).

En satsning i en organisation på modell, metod och verktyg innebär ofta stora investeringar. Inte bara i inköp av t ex metodhandböcker och verktyg, utan också i omfattande utbildning. Man måste investera i personalens kunnskap av metoder och verktyg. En omfattande investering i metoder/verktyg innebär givetvis hinder för nya investeringar. En metodinvestering är "konserverande". Man måste ha goda skäl för att göra nya investeringar och satsa på att människor ska lära nytt och därmed avlära gammalt beteende.

Samtidigt pressar den ständigt pågående teknik- och kunskapsutvecklingen på. Nya insikter, nya metoder och nya verktyg ser dagens ljus i det starkt dynamiska SU-området. Det finns fortfarande många problem i systemutveckling och därmed behov av ökat kunnskap. Den "stabila" metodmiljön i en organisation är ständigt utsatt för ett externt tryck av ny kunskap. Att inte ta till sig nya bättre metoder kostar också pengar. Dataavdelningar är kunskapsorganisationer och som sådana mycket beroende av sin personal, vad gäller kompetens och motivation. Att satsa på kontinuerlig utveckling av metodmiljön kan vara ett konkurrensmedel för att rekrytera och behålla kvalificerad personal.

Det finns inte bara ett behov av standardisering av metodmiljöer. Det är också behov av innovation av metodmiljöer. Det finns ett behov av nytänkande och kontinuerlig kunskapsutveckling om SU-området.

Det finns naturligen en eftersläpning i kunskapen hos många organisationer i relation till kunskapsutveckling inom forskningen, programvaruhus och vissa konsultföretag. Det är inte så att alla teoretiska hugskott är praktiskt nyttiga och användbara. En sund skepticism hos datoranvändande företag är lämpligt att vara försedd med. Genom närheten till sina egna problem sker en kontinuerlig vidareutveckling och anpassning av metodkunskap i många organisationer.

Eftersläpningen i metodkunskap kan faktiskt paradoxalt nog öka genom införande av CASE-verktyg. Många CASE-verktyg är som nämnts standardverktyg för standardmetoder. Man ökar genom användning av sådana verktyg tillämpningen av gamla metoder. Investering i icke-ändringsbara CASE-verktyg kommer att vara starkt konserverande. Att satsa på sådana verktyg innebär enligt min mening ett mycket stort risktagande för en organisation. Man gör sig beroende av en (ofta stor internationell) leverantör och dennes utvecklingsplaner. Går våra utvecklingsbehov ihop med leverantörens kommande utvecklingsarbeten? Man minskar drastiskt möjligheterna till att skapa sig en egen metodprofil. Och då sådana här verktyg dessutom kan innebära en tendens till tankedominans (Goldkuhl, 1991) så ökar riskerna med detta verktygs- och metodberoende.

Systemutvecklingsområdet kommer med stor sannolikhet framledes kännetecknas av en ständig pendling av standardisering och innovation. Det är viktigt att denna växelverkan blir godartad. Vi behöver både ordning och nyskapande. Vi behöver skapa metodmiljöer som är öppna för temporära standardiseringar och successiva innovationer. Vår metodmiljöer måste vara flexibla och förändringsbara; om vi inte tror på det statiska, att kunskapsutvecklingen stannar av.

3.3 SITUATIONSANPASSNING AV METODER

Metoder är standardiseringar av SU-arbete på gott och ont. Varje SU-situation är ej den andra helt lik. Det är skillnader i tillämpningsområde, förutsättningar, problemställningar, tidplaner, ambitionsnivåer och deltagande personer (med kompetens och intresse). Det är närmast en truism att säga att varje SU-projekt är unikt. Men samtidigt finns likheter och det är därför metoder överhuvudtaget är möjliga. Varje SU-situation finns i ett spänningsförhållande mellan det generella och det specifika.

Det har framförts kritik mot metoder för att de är för stela och strikta till sin karaktär. De upplevs ibland som "kokboksaktiga". Den föreskrivna sekvensen av metodsteg känns tung och svårarbetad. I små projekt är den överdriven och känns "byråkratisk". I stora projekt känns den otillräcklig då man inte får stöd för många svåra analysituationer.

De kompetente metoodanvändaren förstår att använda metoden på ett situationsanpassat sätt (Goldkuhl & Röstlinger, 1988). Men detta bygger på att denna metoodanvändare har tillräcklig förståelse för det tänkesätt och förhållningssätt som metoden innebär, och inte bara för metodens tekniska detaljregler (se avs 3.1 ovan).

Situationsanpassning innebär en flexibel, dynamisk och iterativ användning av olika metoddelar. Man är uppgiftsorienterad (mot syftet) snarare än metodorienterad. Vissa delar av metoden kan utelämnas och vissa kan behöva modifieras. Ordningen mellan olika metodsteg kan behöva kastas om. Man kan behöva komplettera med helt andra metoder i den aktuella situationen.

Situationsanpassning är krävande då det behövs god metodkunskap och erfarenhet. Men det är förmodligen nödvändigt om man ska lyckas med metoodanvändning i svåra komplexa SU-situationer. Situationsanpassning blir en naturlig konsekvens av ökat metodkunnande; och där detta professionella metodkunnande även innefattar andra metoder än just de som man för tillfälligt föreskriver i organisationen.

3.4 EGENKOMBINERADE METODPAKET

En historisk utvecklingstendens som redovisades ovan (avs 2.3) var förekomsten SU-modeller marknadsförda av konsultföretag. Vad kommer detta leda till? Kommer varje företag att tillämpa någon av dessa 4-5 modeller? Jag uppfattar att konsultföretagen satsar aktivt i marknadsföring av sina modeller. Det är ett viktigt konkurrensmedel. Det är bl a ett sätt att skaffa sig en dominerande roll som leverantör av konsulttjänster om ett företag tillämpar aktuell konsultmodell.

Konsultmodellerna är starkt standardiserade metoodpaket och de passar kanske inte alla organisationer som kan ha olika och egna metoodtraditioner. Konsultmodellerna innehåller som nämnts ovan ett modellmässigt ramverk och sedan en speciell blandning av olika mer eller mindre etablerade metoder. Just denna förekomst av metoodblandning underlättar på sätt och vis att införa en alternativ metoodblandning. Man kanske byter ut någon metod mot en annan som bättre passar med organisationens arbetssätt. Man har kanske tidigare kunnande i och goda erfarenheter av en viss metod. Man kanske har upptäckt någon viss metod som innebär lösning på återkommande problem i SU-arbete i organisationen. Man har kanske fastnat för något visst synsätt på systemutveckling och önskar utnyttja de metoder som bäst passar detta synsätt. Det kan finnas många skäl till att göra anpassning till organisation.

Om vi återvänder till strukturbeskrivningen (fig 1.1-3) över olika hjälpmedel för systemutveckling: Enligt denna kunde man använda olika metoder i en modellstruktur. Man kan i princip, givet en modellstruktur, byta ut olika metoddelar. Finns det ett moment i strukturen som t ex kallas funktionsanalys kan man kanske använda V-grafer enligt ISAC, MBIs verksamhetsgrafer eller dataflödesdiagram enligt Yourdon. Dessa analys-/beskrivningsmetoder kan uppfattas som ungefärligt ekvivalenta.

Givetvis har de olika egenskaper och därmed kvaliteter. Metodförespråkare uppfattar givetvis stora skillnader och anser dem kanske inte som ekvivalenta. Poängen här är inte att säga att de olika metoderna är likvärdiga, bara att de täcker delvis samma typ av analysuppgift och därmed ur vissa metodanvändares synvinkel uppfattas som utbytbara.

En användare av LOGIC finner kanske att denna metoddel av modellen är otillfredställande; istället för Yourdons dataflödesdiagram väljes t ex ISACs eller MBIs verksamhetsanalys medan andra metoddelar av modellen behålles intakta.

Ett ökat metodkunnande kommer säkert leda till en ökad metodanpassning av använda standardmodeller för systemutveckling. Man kommer att i högre grad att etablera egna kombinationer av metoder.

Låt mig ta exempel från egen erfarenhet som metodkonsult för att ytterligare illustrera detta. Jag har arbetat med metodutbildning och metodinförande av förändringsanalys enligt SIM-metoden. Denna metod baseras på en viss struktur av områden, dvs med här använd terminologi en modellstruktur. Det är fem olika områden som står i visst strukturellt förhållande till varandra. Ett av dessa områden är verksamhetsanalys. För FA/SIM-metoden används här sk handlingsgrafer (Goldkuhl, 1990a). Jag har arbetat med företag som tagit metoden "rakt av". Trots att man har erfarenheter av andra verksamhetsanalysmetoder har man önskat byta och istället tillämpa kontextuell analys med handlingsgrafer. Jag har arbetat med andra företag som just pga inarbetad metodtradition velat hålla kvar vid tidigare etablerade metoder (ISAC respektive MBI). Det är fullt möjligt att tillämpa huvudprinciperna, modellstrukturen och övriga metoddelar i FA/SIM tillsammans med en annan metod för verksamhetsanalys.

Det generella i ovanstående berättelse - att byta ut och kombinera olika metoder - är inte ovanligt idag tror jag. Jag dristar mig (baserat på tidigare resonemang i denna rapport) att påstå att detta kommer att bli allt vanligare i framtiden.

Det finns dock några viktiga förutsättningar. Man måste ha en flexibel modellstruktur, dvs ett ramverk som tillåter utbyte av metoder. Man måste var medveten om de mål som man anser ska gälla för organisationens SU-arbete. Vilket synsätt ska gälla? Det är viktigt att låta synsättet komma först. Alla metoder går inte att blanda hur som helst eftersom de bygger på olika synsätt. Det gäller att etablera fruktbara konceptualiseringar och inspirerande mål och visioner för systemutveckling. Detta är, återigen, grunden för hantering av modeller, metoder och verktyg.

3.5 ANPASSNING AV CASE-VERKTYG

Att användningen av CASE-verktyg kommer att öka är nog tämligen riskfritt att hävda. Men vilken typ av verktyg är det som kommer att öka? Är det standardverktygen med amerikanska standardmetoder som kommer att fortsätta att dominera

marknaden? Det beror delvis på vilka metoder som kommer att dominera i användning. Vissa konsultmodeller bygger ju i stor utsträckning på etablerade metoder och detta kan då innebära att standardverktyg för dessa metoder får en fortsatt stark ställning. LOGIC-modellen innehåller ju bland annat data-modellering och dataflödesdiagram enligt Yourdon. Det är därför möjligt att använda vissa standardverktyg som stöd till LOGIC vilket också görs (verktyget IEW).

Det finns emellertid en risk med detta. Standardverktyg (som IEW) täcker givetvis inte alla metoddelar som LOGIC innehåller. Vad händer med de delar som inte täcks av verktyget? Blir de styvmoderligt behandlade i praktisk tillämpning? Blir de bortglömda då verktyget kanske är mer tankedominant än befintlig metodbeskrivning? Detta är ett resonemang som givetvis gäller generellt, inte bara för kombinationen LOGIC-IEW. Vad händer när man använder verktyg som bara täcker vissa delar av använd modell/metod.

Jag tror också att det finns en frestelse att inkorporera metoder i sin modell som det finns CASE-verktyg för, även om dessa metoder inte är lika bra som andra.

Den ökade flexibilitet i metodanvändning som jag förutskickat (i avs 3.1-3 ovan) behöver som stöd en flexibilitet på verktygsnivå. Detta innebär ett ökat intresse för CASE-skal. Genom CASE-skal blir det överkomligt att "snickra" till sitt eget företagsspecifika CASE-verktyg som passar just den uppsättning av metoder som man tillämpar.

Det kan finnas flera motiv för att välja ett CASE-skal som utgångspunkt för sitt verktygsarbete:

- Man har inom företaget en "allmän" eller egenutvecklad metod som det saknas CASE-verktyg för.
- Man har gjort egna välmotiverade metoदानpassningar av en etablerad CASE-stödd metod.
- Det finns inom organisationen olika lokala varianter av en annars organisationsgemensam metoduppsättning. Det behövs anpassade CASE-verktyg för att stödja dessa olika lokala varianter.
- Man har inom företaget utformat sitt eget "metodpaket" bestående av olika delar från olika kända metoder. Sådana metoदानpassningar förefaller bli allt vanligare och detta innebär givetvis att det för sådana, ofta unika, metodkombinationer saknas färdiga CASE-produkter på marknaden.
- Man önskar ett verktyg som man successivt kan vidareutveckla på basis av gjorda erfarenheter och ökat kunnande om användning av metod och verktyg. Man önskar tillämpa en evolutionär systemutveckling avseende använt CASE-verktyg.
- Eller framför allt så ser man att det kommer att ske en fortsatt kunskapsutveckling kring systemutvecklingsmetoder och man önskar inte bli beroende av vilka vidareutvecklingar som sker av vald CASE-leverantör.

Det finns som sagt flera goda skäl att välja ett CASE-skal som utgångspunkt för arbete med att införa CASE-verktyg i sin organisation.

23

I det dynamiska SU-området behöver vi verktyg som tillåter ändring och successiv vidareutveckling av metoder, dvs vi behöver CASE-skal och med det en ökad kunskap om dess användning. Vi bör vara försiktiga med verktyg som låser fast metodanvändningen.

3.6 AVSLUTNING

Jag har beskrivit några olika tänkbara utvecklingstendenser ovan. Det kommer givetvis finnas andra delvis motverkande eller kompletterande. Vad som är viktigt att se är att vi inte är fångna av utvecklingen. Vi är beroende av den allmänna utvecklingen inom SU-området, men vi är också, i större eller mindre grad, medaktörer och skapare av utvecklingen.

Jag har i min genomgång fokuserat på aspekter kring modeller, metoder och verktyg. Det finns andra omgivande utvecklingstendenser som kommer ha stor betydelse för framtida hantering och utveckling av metodmiljöer. Jag vill här avslutningsvis nämna två sådana faktorer:

Betydelsen av frågor kring systemarkitektur och systeminteraktion kommer säkert att öka (se t ex Magoulas & Pessi, 1991). Den kontinuerliga och växande föränderligheten av system och verksamhet kommer också spela stor roll för framtida metodmiljöer.

LITTERATURREFERENSER

Andersen ES (1991) Systemutveckling - principer, metoder och tekniker, Studentlitteratur, Lund

Anderssen N E m fl (1986) Professionel systemudvikling. Erfaringer, muligheder og handling, Teknisk Forslag, Köpenhamn.

Axelsson, L, Ortman, L (1985) Utvecklingshandboken SVEA - Modell och metoder för administrativt utvecklingsarbete, Kommandata, Stockholm

Axelsson, L, Ortman, L (1990) Direct-modellen - en utvecklingshandbok, Studentlitteratur, Lund

Bergsten P, Bubenko J, Dahl R, Gustafsson MR, Johansson L-Å (1989) RAMATIC - a CASE shell for implementation of specific CASE tools, SISU, Kista

Boström B, Nilsson A, Selldén J (1986) Systemering med datorstöd, Esselte Studium, Stockholm

Bubenko, J (1989) Selecting a strategy for computer-aided software engineering, SYSLAB, Stockholms universitet

Dahlbom B, Mathiassen L (1991) Struggling with quality. The Philosophy of developing computer systems, Inst f informationsbehandling, GU/CTH

Dahlgren, H, Gustafsson M, Johansson L-Å (1991) Modelleringsansatser för begrepps- och datamodellering. Beskrivning och försök till jämförelse, SISU, Kista

Fåhraeus E (1986) Metodkedjornas och verktygens roll vid systemutveckling, Riksdataböndet, Stockholm

Gane CP, Sarson T (1979) Structured systems analysis: Tools and techniques, Prentice-Hall, Englewood Cliffs

Goldkuhl G (1984) ISAC omvärderad, ingår i Nissen HE (1984)

Goldkuhl G (1986) Kunskapsutveckling med kvalitet - några viktiga förutsättningar, IDA, Linköpings Universitet

Goldkuhl G (1989) Datasystem och verksamhetsutveckling, IDA, Linköpings Universitet

Goldkuhl G (1990a) Kontextuell verksamhetsanalys med handlingsgrafer, IDA, Linköpings Universitet

Goldkuhl G (1990b) SIM-metoden för verksamhetsinriktad systemutveckling, NordDATA90, Göteborg

Goldkuhl G (1991) Projektbeskrivning: Principer för metodanpassning av CASE-verktyg (PRIMCASE), IDA, Universitetet i Linköping

- Goldkuhl G, Röstlinger A (1988) Förändringsanalys - Arbetsmetodik och förhållningssätt för goda förändringsbeslut, Studentlitteratur, Lund
- Goldkuhl G, Röstlinger A (1991) VITS - beskrivning av ett FOU-program, Institutionen för datavetenskap, Universitetet i Linköping
- Griethuysen, JJ (Ed, 1982) Concepts and terminology for the conceptual schema and the information base, ISO/TC97/SC5N695, America National Standards Institute
- Hirschheim R, Klein H (1989) Four paradigms of information systems development, CACM vol 32 nr 10
- Hugosson M-Å, Hesselmark O, Grubbström A (1983) MBI-metoden. En metod för verksamhetsanalys, Studentlitteratur, Lund
- Hällström M (1989) CASE - ett verkligt generationsskifte, Svenska PC World, nr 10 1989
- Ingevaldsson L (1985) JSD-metoden för systemutveckling, Studentlitteratur, Lund
- Kall C-O (1984) Motverkar conceptuell datamodellering decentraliserade organisationsformer? ingår i Nissen HE (1984)
- Langefors B (1966) Theoretical analysis of information systems, Studentlitteratur, Lund
- Lie-Nielsen S, Colliander L (1983) Principer för en ny generation systemutvecklingsverktyg, Riksdataförbundet, Stockholm
- Lindholm S (1979) Vetenskap, verklighet och paradigm, AWE/Gebers, Stockholm
- Lundeberg M, Goldkuhl G, Nilsson A (1978) Systemering, Studentlitteratur, Lund
- Lyytinen, K (1983) Reality modeling considered harmful - a need for linguistic framework, ingår i Bubenko (Ed, 1983) Information modeling, Studentlitteratur, Lund
- Lyytinen, K, Smolander, K, Tahvanainen, V-P (1989) Modelling CASE environments in systems development, CASE89, SISU, Kista
- Magoulas, T (1990) Objektmodellens praktiska begränsningar, ingår i Magoulas, Pessi (1991)
- Magoulas, T, Pessi, K (1991) En studie av informationssystemarkitekturer, Inst f informationsbehandling ADB, CTH/GU, Göteborg
- Malmborg L (1991) Software development approaches in CASE-tools, In Proceedings of 14th IRIS (Information systems Research In Scandinavia), Umeå

- Mumford E, Henshall D (1983) Designing participatively. A participative approach to computer systems design, Manchester Business School
- Nilsson AG (1991) Anskaffning av standardsystem för att utveckla verksamheter, Handelshögskolan, Stockholm
- Nissen HE (Red, 1984) Systemutveckling - av vem, för vem och hur? Arbetarskyddsfonden, Stockholm
- Nurminen M (1988) People or computers: Three ways of looking at information systems, Studentlitteratur, Lund
- Persson S (1977) Systemskisser, Handelshögskolan, Stockholm
- Selldén J (1988) Systemeringsmetoder inför 90-talet. Ett sätt att forma utvecklingskulturen, Intention, Linköping
- SIS (1973) Riktlinjer för administrativ systemutveckling, Handbok 113, Sveriges Standardiseringskommission, Stockholm
- SIS (1989) Referensmodell för systemutveckling, Teknisk rapport 321, Sveriges Standardiseringskommission, Stockholm
- Sundgren B (1981) Databaser och datamodeller, Studentlitteratur, Lund
- Swende, E (1982) Introduktion av IRM-koncept på SAS, NordDATA82, Göteborg
- Swende, E (1989) Data-driven strukturering, SVING & SVIG-Konferens, CTH, Göteborg
- Söderström P, Olstedt D (1988) Moderna systemutvecklingsverktyg, Riksdataförbundet, Stockholm
- Söderström P m fl (1990) CASE. En plattform för 90-talets Systemutveckling, Dataföreningens Förlag, Stockholm
- Targama A (1981) Att genomföra administrativa förändringar, Sveriges Mekanförbund, Stockholm
- Utvecklingsprogrammet (1988) Medverkan i utveckling, Arbetsmiljöfonden, Stockholm
- Yourdon E (1989) Modern structured analysis, Prentice-Hall, Englewood Cliffs