

# 4. generasjons systemutviklingsverktøy

AV SVEIN A. LØKEN



*Siviløkonom Svein A. Løken har internasjonal konsulentpraksis og har i mange år forelest ved norske høyskoler. Han er for tiden tilknyttet Bedriftsøkonomisk Institutt som studierektor i informasjonssystemer og databehandling.*

## 1. Informasjonsverdi

Det er alment akseptert at det viktigste problem foretakene står overfor i sin EDB-anvendelse er den for lave produktivitet i systemutviklingsarbeidet. En rekke intervjuundersøkelser har vist dette. (1)

En økning i produktiviteten kan oppnås både ved å lage bedre EDB-baserte styringssystemer og ved å gjøre dette med en lavere ressursinnsats. I mange foretak presses det nå på for å ta i bruk nye hjelpemidler i systemutviklingen. Dette skjer ikke uten virkninger for revisors metoder og muligheter.

AAAs definisjon av lovbefalt revisjon beskriver en bevisopptagning for presisjonen i utsagn om et foretaks økonomiske hendelser og tiltak. For den videre diskusjon i denne artikkelen kan det være nyttig å splitte opp og forstørre deler av denne definisjonen.

Hendelser i objektsystemet (omverdenen) avbildes ifølge modeller (f.eks. konteringsregler) som data. Dataene manipuleres, lagres og transporteres av datasystemet. En bruker tolker ifølge sin tolkningsramme et utsnitt av

dataene til informasjon. Denne informasjonen brukes til å redusere usikkerhet i en beslutning.

Modellene kan komme f.eks. fra regnskapsteorien. Metodene for databehandlingen kommer fra teoridannelsen innen systemeringsfaget. I praksis ser vi at systembyggerne langt på vei lar sin virkelighetsoppfatning dominere. Dette skjer delvis på bekostning av regnskapsteorien, men i større grad på bekostning av faktorer vi kjenner som sentrale innen internkontrollen.

Informasjonens verdi kan betraktes som avledet fra de økonomiske konsekvenser en beslutning får. (2) Uten å gå inn på dette i detalj, vil vel de fleste kunne slutte seg til at informasjonsverdien stiger når informasjonen:

1. Er informasjon og ikke bare data (dvs. den kan tolkes)
2. Er relevant, dvs. den påvirker beslutningen
3. Kommer i tide
4. Er eksklusiv
5. Har kjent fullstendighet og pålitelighet.

De to første punktene er sentrale for en utvikling vi kan observere mot brukerstyrt systemutvikling. Det tredje punktet er en sentral årsak til utviklingen mot sanntids datasystemer. Det fjerde punktet er den egentlige begrunnelse for databransjens nest sterkeste motebølge for tiden: data-sikkerhet. Det femte punktet er det dessverre få som interesserer seg for.

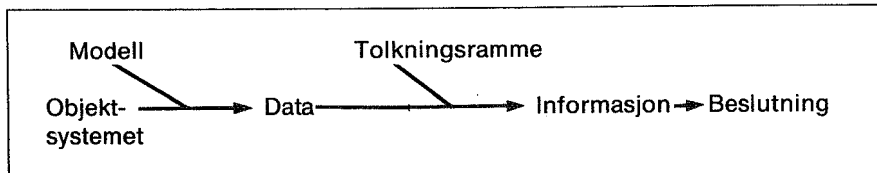
Derfor skal jeg i resten av artikkelen gi momenter til en diskusjon av dette i lys av den utvikling vi ser av EDB-verktøy.

## 2. Validering

Gitt at vi har valgt en modell for å avbilde virkeligheten, må vi danne oss en mening om i hvilken grad registreringen av data følger modellens regler, m.a.o. må vi utforme metoder for dataregistrering (3) og foreta en systembekreftelse (compliance test). (4)

Det er grunn til kort å nevne de forskjellige metoder for datavalidering fordi nyere EDB-systemer dessverre ser ut til å gi dårligere input-kontroll enn eldre, satsvise datasystemer.

1. Eksistenskontroll  
Gyldig kontonr., reskontrnr., etc. ifølge oppslag mot register.
2. Naturgitt eksistens  
Eks.: Dato med mnd. < 12, etc.



3. Numerisk/alfanumerisk felt.
4. Sekvensavhengige felt  
Eks.: Felt for reskontrnr. er kun gyldig dersom hovedbokskonto er definert med hovedkontro.
5. Obligatoriske/valgfrie felt.
6. Kontrollsiffer.
7. Sekvensnummererte poster.
8. Kontrolltotaler (naturlige og nonsens-totaler).
9. Rimelighetskontroller.
10. Fulltekst verifikasjon av innslått kode  
Eks.: Operatør taster kundnr., systemet svarer med navn, adresse.
11. Dobbelregistrering  
(Er sjelden kostnadseffektivt).

Det er grunn til å kommentere tre av disse.

Kontrollsiffer var en svært hyppig anvendt metode for å sikre korrekt registrering av nøkkelbegreper i eldre, satsvise systemer. Ved innføringen av terminalorienterte systemer fjernes ofte denne kontrollen fordi man erstatter den med andre, f.eks. fulltekst verifikasjon av innslått kode. Dette er beklagelig fordi metoden tok hensyn til erfaringsdato for hvilke feil som oftest forekommer, bl.a. omkast av to sifre og dobbeltslag av ett siffer. Selv om vi ikke lenger finner det nødvendig å gjennomføre utregningen av kontrollsifferet ved transaksjonsregistrering fordi vi f.eks. kontrollerer eksistens av kundnr. ved oppslag mot register, vil det være en åpenbar styrke å tildele gyldige kundnr. som inneholder kontrollsiffer. Forbytningsrisikoen reduseres sterkt. (5)

Mange systemkonstruktører legger stor vekt på fulltekst verifikasjon av innslått kode, og denne metoden brukes ofte til erstatning for kontrollsiffer. Legg da bare merke til at metoden forutsetter at operatøren aktivt reagerer på den tilbakemeldte informasjonen. EDB-systemet har en passiv kontrollfunksjon. Mange forsøk har vist hvor svak en slik kontroll kan være. La meg nevne tre faktorer.

1. Ved hyppig anvendelse reagerer operatøren ikke på svaret.
2. I systemer med liten feilrate reagerer ikke operatøren på svaret.
3. I pressede situasjoner (da feilraten kan tenkes å være størst) reagerer operatøren ikke på svaret.

Den valideringsmetoden bør derfor reserveres for situasjoner der det er vanskelig å anvis andre, aktive kontrollmetoder.

Det har vist seg vanskelig å få systemkonstruktører til å anvende rimelighetskontroller. Det kan skyldes at det er ressurskrevende å etablere intervaller for rimelige verdier. Det er derfor meget beklagelig at systemkonstruktører ikke anvender de samme metoder i inputkontrollen som flere revisjonprogrammer har anvendt a posteriori i lang tid: nemlig å basere seg på løpende oppdaterte statistiske fordelinger. Det fører for langt å gå inn på dette i detalj i denne artikkelen, men la meg gi en idé om hvordan man kunne gå frem.

I tilknytning til hver konto finnes et felt som løpende oppdateres med gjennomsnittlig beløpsstørrelse (absolutte tall) for posteringer på kontoen. Antar vi en poisson-fordeling av beløpene, kan vi dermed gi en grense vi kan teste mot for å finne beløp som er f.eks. mindre enn 4 % sannsynlige.

Hovedpoenget er at denne testen kan utformes generelt (for beløp, mengde, etc.) slik at kostnaden ved å inkludere den som en standard inputkontroll blir lav.

### 3. Tolkning

Rapportens overskrifter og brukerens systemkunnskap er sentrale faktorer når informasjonsbrukeren skal tolke en rapport. Revisjonssporet skal gjøre det mulig å verifisere at rapportinnholdet er en korrekt avbildning av et sett hendelser målt ved grunndata.

I nyere EDB-systemer har vi sett behov for data om rapporters utsegnskraft. La meg angi tre typiske områder.

I beslutningsstøttesystemer (6) kan det være aktuelt å registrere antagelser sammen med observerte data. Vi registrerer kanskje et ekspertpanels forventninger om et totalmarkeds bevegelse. Når vi siden konstruerer rapporter hvor denne type data kombineres med historisk registrerte data, vil mange føle behov for å se hvor mye av «rapportens konklusjon» som stammer fra anslag og hvor mye som kommer fra målinger.

I systemer som kombinerer regnskapsdata med tekniske data er det kjent at man må behandle konsekvenser av målefeil (måleuøyaktighet) med omtanke.

Jeg har nær sagt at i takt med skiftende økonomidirektører ser man at lay-out for regnskapsposteringene endres. Nye hjelpemidler i systemutviklingen gjør det langt billigere enn før å endre utformingen av et EDB-basert regnskapssystem. Jeg kommer tilbake til dette i neste hovedavsnitt.

Over tid vil nye registreringskriterier komme til og gamle falle ut. Et problem er det i dagens systemer at vi sjelden gir oss selv mulighet til å skille mellom «ikke-observasjon» og verdien null. Det er da klart at gjennomsnittsberegninger, trendutvikling o.l. lett kan bli meningsløse. Vi som utformer økonomiske styringssystemer vil ha mye å lære av å se hvordan f.eks. statistisk sentralbyrå behandler dette problemet.

### 4. Nye hjelpemidler i systemutviklingen

For å bøte på det edb-bransjen kjenner som sitt viktigste problem, systemutviklingsproduktiviteten, tilbys det i markedet det som kalles 4. generasjonsspråk. (7) Det er programmeringsverktøy og systembeskrivelsesverktøy som bedrer produktiviteten kanskje 1:20. (8)

Verktøyene er mye enklere å bruke enn tradisjonelle programmeringsspråk, og dette fører til at brukerne selv kan definere sine systemer. De kan raskt konstruere skjermbilder for dataregistrering og rapporter for datapresentasjon.

En kjerne i alle disse verktøyene er data dictionary eller spesifikasjonsdatabasen (SDB). Noen kaller den også for metadatabasen.

Her beskrives alle data et EDB-system skal behandle.

Beskrivelsen kan f.eks. se slik ut:

```
ELEMENT ktonr ALIAS konto kon-
tonummer PICTURE 9999
HEADING«Ktonnr» LABEL
«Ktonnr»
VALUES 10000 TO 99999
UNIQUE IN kontoplan
LOOKUP ON kontoplan
HELP «Tast kontoangivelse som
femsifret kontonummer»
DESCRIPTION
```

Ktonnr er det begrep vi bruker for å . . . . .

Vi har her beskrevet begrepet ktonr, gyldige verdier for dette begrepet, at kontonummeret må være entydig i kontoplanen, osv.

I de fleste sammenhenger vil vi betrakte det som et stort fremskritt at vi med dette har fått ett sted der både bruker og revisor kan gå for å finne dokumentasjon av systemet.

Det er denne sentraliserte systemdokumentasjonen som danner grunnlag for en automatisering av deler av programmeringsprosessen. Dermed kan vi altså oppnå dramatiske produk-

tivitetsforbedringer. Dette har ført til en endret stil i utviklingsarbeidet der eksperimentering med alternative løsninger delvis overtar for gjennomanalyserte systemforslag. Sannsynligvis innebærer dette en øket mulighet for feil og utilsiktede virkninger av et system.

Tradisjonelt hadde vi en arbeidsdeling mellom bokholderi og kasse. Da de første EDB-baserte regnskaps-systemene kom, forsvant etter hvert denne delingen, men den ble erstattet med en ny: systemer/programmer og bruker. Denne arbeidsdelingen står opplagt i fare for å forsvinne ved innføringen av 4. generasjons system-utviklingsverktøy.

## 4.1 Dataregistrering

En bruker vil kunne referere til elementet ktonr når han bygger et skjerm bilde for dataregistrering. Ved å gjøre dette «vet» systemet noe om begrepet ktonr slik at f.eks. eksistenskontrollen kan foretas automatisk.

Å konstruere et skjerm bilde som gjør det mulig å registrere en regnskapstransaksjon tar typisk mindre enn 30 sekunder. Kommandoene kan f.eks. være disse:

```
SCREEN transregistrering
FILE poster
GENERATE
BUILD
```

Skjerm bildet kan så benyttes til å registrere en eller flere transaksjoner, for deretter å forsvinne. Motforestillningene gir seg selv.

Det er viktig å være klar over at de aller fleste verktøy av denne type legger begrenset vekt på autorisasjon og databeskyttelse og enda mindre vekt på datavalidering. Det er ikke unaturlig tatt i betraktning hvorfor og hvor de er utviklet. Flere av de mest anvendte verktøy har egenskaper som til og med gjør det svært kostbart å anvende de metodene for datavalidering som ble nevnt i et tidligere avsnitt. Jeg kjenner eksempler der en generell produktivetsforbedring 1:20 avløses av en produktivetsreduksjon 5:1 når det gjelder å lage effektive og sikre dataregistreringsrutiner.

For å muliggjøre revisors vurdering av internkontrollen, må vi stille disse krav til de nye verktøyene:

1. Datavalideringen må kunne beskrives i spesifikasjonsdatabasen.
2. Datavalideringen bør ikke kunne spesifiseres i det enkelte skjerm bilde.

Få andre enn revisor er opptatt av slike krav.

## 4.2 Korrumpert av lagrede data

Det er ikke plass i denne artikkelen for en diskusjon av autorisasjon. Jeg bare minner om at innføringen av 4. generasjons programmeringsspråk forsterker behovet for gode metoder for autorisasjon av systembrukerne.

Uansett vil behovet for å kontrollere på annen måte at data ikke er endret forsterke seg. Mange vil finne nye teknikker for avstemminger nyttige. En slik teknikk kalles polynomsum-avstemming.

**Problemet:** Hvordan sikre oss at en post i tidligere (revidert) periode ikke endres?

**Mulig løsning:** I tilknytning til hver post defineres et dataelement som beregnes slik: Hvert felt multipliseres med en vekt. Produktene summeres. Produktsummen lagres i kontrollfeltet. Det finnes da teori for hvorledes disse vektene skal konstrueres for å gi minst mulighet for endring av feltene uten at dette vises i kontrollfeltet. (Forklaringen ovenfor viser prinsippet, i praksis vil feltene være faste 16, 32 eller 64 bits felt og kontrollfeltet inneholde resten ved summen dividert med et fast tall.) Ved oppsummering til varierende tidspunkter vil en sum av kontrollfeltene kombinert med en telling av postene da vise at postene ikke er endret.

Denne metoden har en klar parallell i hva som gjøres for å beskytte data-transport mot korrumpert. Interesserte lesere henvises til de nylig vedtatte standarder for beskyttelse av pengeoverføringer, the Financial Institution Message Authentication Standard (FIMAS) X9.9. Da denne standarden også er en standard for kryptering, må den benyttes sammen med standard ANSI X9.17 Key Management Standard. (9)

## 4.3 Verdireduksjon gjennom uautorisert spredning av data

Som tidligere nevnt er datasikring et moteord i EDB-verdenen. Legg merke til at flere konsulentfirmaer er startet med datasikring som arbeidsfelt, og at Den Norske Dataforening har opprettet interesse- og studiegrupper i tilknytning til temaet.

La meg minne om at målet er å sikre høy informasjonverdi. På denne bakgrunn ønsker vi en best mulig spred-

ning av informasjon innen en beslutningsgruppe eller et foretak, og en minst mulig spredning utenom beslutningsgruppen eller foretaket. De nye programmeringshjelpemidler gjør spredning enklere, men forbedrer ikke uten videre kontrollen med denne spredningen.

## 5. Konkluderende bemerkninger

Produktivetsforbedringen vi oppnår ved å ta i bruk 4. generasjons system-utviklingsverktøy er så formidabel at vi må forvente at verktøyet får innpass i de fleste organisasjoner. Det vil gi oss nye muligheter til løpende å tilpasse økonomistyringssystemene til skiftende formål, noe vel de fleste vil applaudere.

Probleme med å sikre en tilfredsstillende internkontroll blir større. Det blir også vanskeligere å vurdere kvaliteten av internkontrollen. Disse to problemer kan vi avhjelpe ved å velge verktøy der god internkontroll og revisorbarhet også har vært designkriterier for de som har utviklet de nye hjelpemidlene. Det gjelder dessverre for mindre enn 5% av de markedsførte systemene.

Min tro er at de momenter jeg har drøftet i denne artikkelen normalt ikke tas med i vurderingen når bedriftene bestemmer seg for ett 4. generasjons-språk. Derfor er det nødvendig at revisor påtar seg jobben med å få disse kriteriene med i vurderingen. Mitt håp er at så vil skje.

### LITTERATUR:

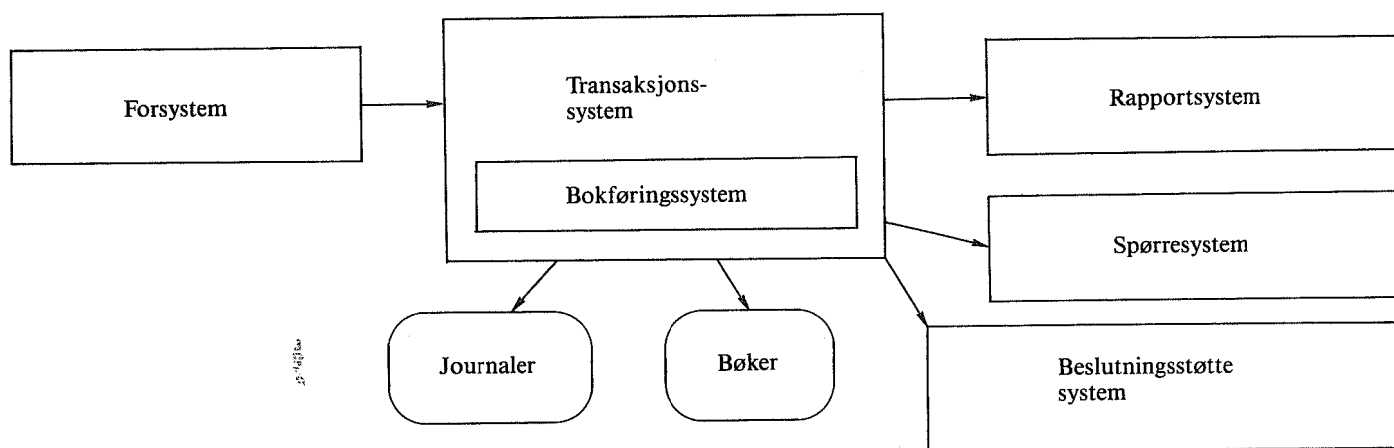
- (1). Pitchell, J. J., «Presidents Report», GUIDE Quarterly Report 3, Sept. 1982.
- (2). Marchak, Jacob og Roy Radner, Economic Theory of Teams, London 1972.
- (3). Davis, Gordon B., Computers and Information Processing, Tokyo 1978.
- (4). Weber, Ron, EDP Auditing, Conceptual Foundations and Practice, Tokyo 1982.
- (5). Daniels, Alan og Donald Yates, Systems Analysis (Palo Alto Science Research Associates, Inc. 1971).
- (6). Stabell, Charles, Decision Supporting with Spreadsheet Programs, foreløbig utgave B. I., sept. 1985.
- (7). Martin, James, Application Development without Programmers, Prentice Hall, 1982.
- (8). Rudolph, Eberhard E., Productivity in Computer Application Development, Working Paper no 9, The University of Auckland, N. Z., 1983.
- (9). Neuman, Morris, Financial Users can bank on new security standard X9.17, artikkel i Data Communications, Juni 1985.

# Svein A. Løken: Økonomistyringssystemet – er det godt nok?

*Er du ansvarlig for oppbygning av bedriftens økonomiske styringssystemer? I så fall drar du kanskje kjensel på fire situasjoner som er grunnlaget for denne artikkelen. Jeg har møtt dem flere ganger de siste par årene, og tror de er typiske.*

## Avgrensning

Ofte lager man et bilde av regnskapsinformasjonssystemet og tilstøtende systemer slik:



Forsystemer er systemer som leverer data på maskinlesbar form til regnskapsinformasjonssystemet. Et eksempel på et forsystem er transaksjonsoverføring mellom bank og bedrift. Journaler og bøker innbefatter eksternregnskapets lovbefalte utskrifter, mens internregnskapets rapporter og saksbehandlers informasjonsbehov dekkes av rapportsystemet og spørresystemet. Forholdet mellom begrepene transaksjonssystem og bokføringssystem er uklart. Ikke sjelden benyttes «transaksjonssystem» av forfattere med bakgrunn i databehandling der forfattere av regnskapsbøker benytter «bokføringssystem». (Se f.eks. Rahman, M/Halladay, M.: Accounting Information Systems, Prentice Hall, N. J., 1988). Personlig foretrekker jeg å la termen transaksjonssystem inneholde bokføringssystem, slik

at transaksjonssystem i tillegg innbefatter datatekniske opplysninger om realiseringen av bokføringssystemet (på en datamaskin ved hjelp av programverktøy). I denne artikkelen skal jeg i hovedsak holde meg til bokføringssystemet.

## Typiske kontostrukturer

Grunnlaget for de analyser som kan foretas i regnskapet er konteringen, det vi også kunne kalle økonomisk klassifikasjon av hendelser i bedriften. Det er to hovedprinsipper som konkurrerer, hierarkisk klassifikasjon og flerdimensjonal klassifikasjon. Hierarkisk klassifikasjon er den eldste; systematisk inndeling av (hoved-)konti i underkonti har vi hatt ihvertfall i femti år. Strukturen er enkel å forstå, og gir enkle behandlingsregler ved regnskapsavslutning og

rapportering. Generalkontoplanen bygger vesentlig på denne tankegangen. Svært mange norske regnskapssystemer ligger nær opp til denne ideen.

Efterhvert begynte nok mange å føle den hierarkiske inndelingen som en tvangstrøye, og utviklingen ga oss reskontri, budsjettansvar og prosjektregnskap. Ihvertfall de to siste representerer et brudd med den tradisjonelle inndelingen i konti og underkonti. Karakteristisk for ansvarsregnskapet er at økonomiske hendelser grupperes uavhengig av hovedkontienes systematikk, eller mer presist: vi har fått et grupperingsbegrep i tillegg til konti og underkonti. Prosjektregnskapet bryter med øvrige behandlingsregler ved at det innfører en annen tidsavgrensning enn hovedregnskapet. For de interesserte er det hyggelig å kunne henvise til en dansk bok fra sekstitallet, nemlig Vagn Madsens lille «Oplysnings- og Variabilitetsregnskap».

Dette at vi konterer med flere begreper enn konto, leder oss over i en *flerdimensjonal* kontostruktur. Ordet flerdimensjonal henspiller på at vi får muligheten til å se på regnskapssummer på alternative måter. Kontovis oppsummering er ikke lenger eneste mulighet. (En parallell til dette finner vi i bibliotekssystemer og tegningsarkiver: i tillegg til hierarkisk desimalklassifisering har man fått gjenfinning etter emneord). Det er nær sammenheng mellom økningen i antall dimensjoner og tilgangen til databehandlingsmaskiner: Det var teknologien som gjorde endringen praktisk mulig.

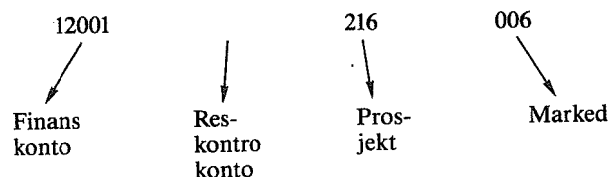
### Kostostreng kontra dataelement

Ved hierarkisk klassifisering registrerer vi konteringen i en kostostreng: en sammenhengende rekke med siffer (evt. iblandet bokstaver). Kostostrengen har gjerne en fast inndeling i deler; hovedkonto, underkonto, e.l.



Karakteristisk her er at de enkelte kostostrengdeler ikke har noen fast betydning, underkonto 01 kan på en konto bety varebiler, på en annen konto bety provisjon til egne selgere (dette eksempelet følger ikke NS4102). Denne ordningen av kontosystematikk fungerer bra ved små systemer og også i store systemer der klassifiseringen er statisk. Imidlertid har mange

etterhvert møtt store problemer der strukturen bryter sammen («det er ikke flere ledige siffer»). Kostostrengorienterte systemer utvides gjerne med kostostrengdeler (felter) med fast betydning, så som reskontrnr, avdelingsnr, o.l. Vi nærmer oss da den generelle ordning som kalles dataelementinndeling av klassifikasjonen.



Det karakteristiske er:

I utgangspunktet er alle deler likeverdige, ingen felt er overordnet, selv om det kan eksistere avhengigheter mellom feltene.

Ikke alle felt må fylles ut. (reskontrofeltet i eksempelet ovenfor er blankt)

Et felt har en fast betydning.

Hvert felt vil representere en dimensjon i regnskapet, en ny mulig måte å trekke sammen posteringer på. Denne organiseringen er nært beslektet med databehandlings såkalte 4. generasjons verktøy, og brukerne møter gjerne denne inndelingen i forbindelse med regnskapssystemer som er koblet sammen med en rapportgenerator.

Det er mye diskusjon om å knytte faste verdier (typer) til feltverdier, og det fremheves gjerne av selgerne av de mest solgte større systemer på markedet som viktige fortrinn og nyskapninger. Et godt eksempel burde være ønsket om å klassifisere balansen slik vi gjør i regnskapsanalysen: Mest likvide aktiva, mindre likvide aktiva, anleggskapital, osv. Vi kunne gjøre dette ved å registrere en kode for likvidgrad når vi posterer på kontoen, men bedre ville det være å knytte en fast kode for likvidgrad til kontoen. Dette kan vi gjøre fordi alle posteringen på en bestemt konto alltid har samme likvidgrad. Vi kunne også tenke oss at postering på et bestemt prosjekt og en bestemt konto alltid skulle vedrøre et bestemt budsjettansvar. Da kunne vi ha regler i bokføringssystemet som automatisk skapte korrekt kode for budsjettansvar uten at operatøren behøvde å taste budsjettansvaret.

For regnskapets informasjonsinnhold er dette uinteressant. Den diskusjonen hører hjemme sammen med annen diskusjon om den datatekniske realisering av systemet, dvs. det som datafolk kaller

normalisering av datamodellen. For det praktiske opplegget av registreringen i bokføringssystemet kan imidlertid slike teknikker for å redusere omfanget av konteringsarbeidet være viktig.

(Spesielt datainteresserte vil kjenne igjen feltene som *egenskaper* i en datamodell dersom vi betrakter den økonomiske hendelsen som entitet).

Vi kan observere en overgang fra klassifikasjon ved hjelp av kontostreng til en klassifikasjon ved hjelp av uavhengige dataelementer. Poenget ved denne måten å klassifisere på er at vi enkelt kan aggregere data på mange ulike måter, og det skal ikke mange felt tilk før antall mulige kombinasjoner blir stort. Dette leder til en ny utvikling av regnskapssystemene: reduksjon av antall lagrede saldi eller fjerning av saldi helt fra databasen.

### Saldoorientert kontra saldoløst system?

Alle bokføringssystemer må innbefatte mulighet til å registrere saldi (i det minste IB på konti). Så i denne betydningen finnes det ingen saldoløse systemer. Det er så heller ikke det som menes. Med et saldoorientert system mener vi at det i tilknytning til et konteringsbegrep (f.eks. konto) lagres et sumtall i systemet (f.eks. IB for inneværende periode). I et saldoløst system finnes det ingen slike summer lagret, også evt IB fra 1. januar lagres som en transaksjon. Et saldoløst system kan da også kalles et transaksjonsorientert system.

Tre forhold fører til at vi *ikke ønsker å beregne og lagre saldo på konti ved regnskapsavslutning eller ved bilagsregistrering:*

Vi er sjelden interessert i kontosaldo uavhengig av andre felter (f.eks. avdeling). Datasystemet blir mye mer robust når saldi ikke lenger skal lagres. (Differanseer mellom transaksjonsfiler og kontofiler er ikke mulig fordi kontofilen ikke inneholder beløp).

Ved analyse er vanligvis transaksjonene mer verdifulle enn historiske saldi fordi enhver aggregering medfører tap av informasjon.

### Ulempene er også klare:

I systemer med stor transaksjonsmasse (f.eks. mer enn 500.000 transaksjoner pr år) vil datamaskinbelastningen bli stor (høy databehandlingskostnad/lange svartider).

Det stilles nye krav til internkontroll og edb-revisjon, krav som tildels er systemspesifikke og maskinspesifikke.

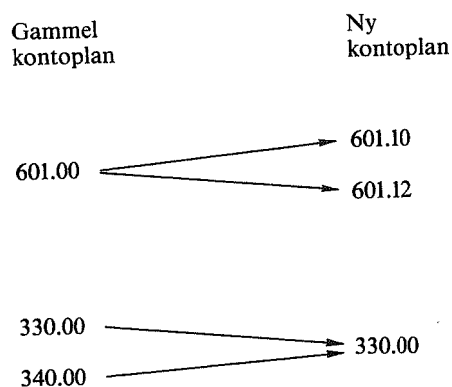
I større systemer opereres det ofte med hjelpesaldi som reduserer behovet for gjennomlesning av transaksjoner. Denne løsningen står og faller med hvor stor del av regnskapsrapportene som er standardisert og repetitive. En annen mulig vei å gå er å spre transaksjonsmassen på flere datamaskiner (f.eks. avdelingsdatamaskiner). Meg bekjent finnes det ikke regnskapssystemer på markedet som baserer seg på programvare som tillater distribuert lagring av transaksjonene.

Men la dette være nok bakgrunnsstoff. Hvilke kanskje nye, kanskje typiske situasjoner har jeg møtt i større bedrifter det siste året når det gjelder oppbygningen av regnskapssystemene?

### Situasjon 1: Den nye økonomidirektøren.

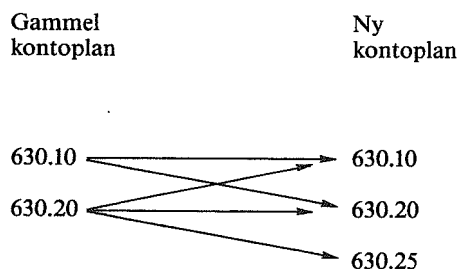
Ved skifte i den økonomiske toppledelsen er det vanlig at man ønsker å endre kontoplaner og rapporteringssystematikk. Og det forløper ikke uten videre smertefritt. De nye konteringsreglene betyr et brudd i mulighetene for analyser med tallmateriale som spenner over tiden før og etter regelskiftet. Hvorfor det oppfattes som problematisk vil du finne noen tanker om ved slutten av artikkelen. La meg oppsummere hva jeg oppfatter som typiske problemer:

Den enkleste situasjonen har man der hvor man legger opp til en ren oppsplitting av en tidligere konto i flere nye, eller samler flere tidligere konti i en ny. Analyser kan da gjennomføres i den kontoplanen som har færrest konti. Teknisk skjer dette i regnskapssystemets rapportgenerator eller (oftere) i et regneark. (De fleste regnskapssystemene har idag hjelpemidler for å eksportere tall maskinelt til regneark).





Mer vanlig, men vanskeligere blir det i dette tilfellet:



Enkelte har forsøkt å etablere forholdstall i regnearket («60 % av de gamle posteringene på 630.10 ville med nye regler havnet på 630.10, 40 % på 630.20»). Forholdstallene kan konstrueres ved å trekke et statistisk utvalg fra transaksjonene kontert etter gamle regler og vurdere virkningen av nye konteringsregler for disse.

En helt annen løsning består i rekontering av eldre transaksjoner. Vi støter her på følgende problemer:

Revisjonsspor fra eldre (allerede produserte rapporter) må ikke brytes.

Internkontrollprosedyrer i nye saldoløse regnskapssystemer beregner en polynomsum pr transaksjon som fremkommer på revisjonsjournaler etc. Poenget med denne summen er nettopp å avdekke at ingen har endret på transaksjoner etter at internrevisjon/eksternrevisjon har revidert den periode der posteringen hører hjemme.

De fleste regnskapssystemer tillater ikke rekontering av gamle transaksjoner. (Riktignok kan man bruke korreksjonsbilag, men det kan være umulig å henføre disse til korrekt periode).

Er verdien av merinformasjonen ved fornyet kontering større enn kostnaden ved konteringsarbeidet?

Eksempelene ovenfor innebærer en omgruppering av konti. Vanligere er det at man legger til en eller flere dimensjoner i regnskapet (ved å definere nye konteringsfelt). Dette ville i eksemplene ovenfor tilsvare en splitting av en konto i flere.

Når analyser over tid foretas av de personer som er vel fortrolige med de endringer som har skjedd i konteringsreglene og som teknisk kan inkludere endringene i analysen, vil fordelene med rekontering begrense seg til økningen i informasjonsverdi som rekonteringen medfører. Alvorligere er det når saksbehandlere benytter beslutningsstøttemodeller uten å være klar over regelendringen. Da kan resultatene svekkes, i verste fall være villedende. Jeg har flere ganger sett eksempler på rapporter med feil i

bedrifter der brukerne benytter rapportgenerator/spørrespråk regelmessig på eldre transaksjonsmasse og konteringsregler har vært endret.

### Situasjon 2: Det nye økonomistyringssystemet

Utskiftingen av regnskapssystemer i de større bedriftene har vært omfattende de siste to årene. Formålet har stort sett vært å få til mer fleksibel rapportering ved å gå over til (i hvert fall tilnærmet) flerdimensjonale saldoløse systemer. Man har begjærlig tatt i bruk muligheten for omfattende klassifisering av økonomiske hendelser i bedriften. Og brukerne har raskt lært å sette pris på forbedrede spørremuligheter og mer omfattende rapportering. Fullt så begeistret er ikke de som konterer. Arbeidsmengden, og dermed kostnaden ved å kontere øker tildels kraftig. Det stilles også større krav til konteringshjelpemidlene og opplæring i kontering. Engelske revisjonsselskaper har også rapportert at konteringsnøyaktigheten reduseres ved lange konteringsstrenger/mange konteringsfelt. Disse forholdene peker ut to nye områder som satsningsområder for systemkonstruktørene:

Regnskapssystemer der det tillates at felt som normalt er obligatoriske ikke fylles ut, f.eks. hvis transaksjonsbeløpet er lavt. Man kan anta at dette allerede skjer i en viss utstrekning dersom datasystemet ikke kontrollerer hvorvidt slike obligatoriske felt faktisk blir utfylt. Skal rapporter og analyser bli korrekte må man definere en ny feltverdi: «ikke klassifisert» eller «ikke utfylt» og sørge for at rapportgeneratorer er istand til å behandle denne verdien. Det er ikke uten videre gitt at denne verdien bør være 0 (null).

For å redusere arbeidsomfanget ved kontering vil det være aktuelt å utvikle et beslutningsstøttesystem integrert med transaksjonsregistreringssystemet. Et slikt system vil muligens også kunne bedre kvaliteten i konteringen, men dette er på ingen måte sikkert. De fleste registreringssystemer tilbyr idag noe informasjon om betydningen av de enkelte felt i konteringsbildet ved hjelp av «Hjelptast», «?», mus og rullegardinmenyer e.l. Bedre ville det være om registreringsbildet dynamisk kunne bringe all den informasjon som vanligvis finnes i konteringsdokumentasjonen og gjøre den tilgjengelig for registrerings- eller konteringspersonalet ved behov.

La meg ta et (litt forenklet) eksempel (fra skisseløsning for nytt brukergrensesnitt for IDE-systemet):

Bedrift: LK		Operatør: SAL		Bunke: 19		Bunkesaldo	123.000.00	65.000.00-
						Bilagssaldo	57.000.00	0.00
Dato	Bilag	Konto	Resk	Ansvar	Dim	Beløp	Tekst	Valkod
12.03	15	42001		115		57.000.00		DEM
12.03	15	43100						

Markøren står i ansvarsfeltet. Hvis operatøren trykker Hjelpstast e.l. vil bildet skifte slik:

Bedrift: LK		Operatør: SAL		Bunke: 19		Bunkesaldo	123.000.00	65.000.00-
						Bilagssaldo	57.000.00	0.00
Dato	Bilag	Konto	Resk	Ansvar	Dim	Beløp	Tekst	Valkod
12.03	15	42001		115		57.000.00		DEM
12.03	15	43100						

Gyldige ansvar er

512 Andersen  
117 Olaisen  
220 Pedersen

Pedersen har ansvaret for kvalitetskontroll ved ekspedisjon av varer. Se organisasjonshåndbok 2.16.110 Budsjet d. konto 100' Rest 70' budsjet totalt 180' Rest 110'

Beslutningsstøtten ligger i at de alternative budsjettansvar er sortert slik at mest sannsynlige alternativ kommer først («512 Andersen»). Operatøren har her bedt om ytterligere opplysninger om ansvar 220. Den praktiske ide er at operatøren får konteringshjelp i suksessivt større dybde ved behov og uten at det opprindelige skjerm-bilde helt forsvinner. Dette kalles *vindusteknikk* og medfører sannsynligvis at man må gå over fra «uintelligente» terminaler til arbeidsstasjoner med lokal databehandlingskapasitet i dataregistreringen.

Det vil føre for langt å an vise hvordan rangering av alternativene her kan skje, men kort fortalt vil rangeringen kunne avhenge av konto, operatør og beløpsstørrelse. Dette vil kunne medføre at rekkefølgen av felter må bli en annen en før: svært ofte blir beløp et av de første feltene vi taster. (Grunnlaget for konteringshjelpen må komme før konterings-elementene). Sannsynligheten for at en slik konteringshjelp er kostnadseffektiv øker med kompleksiteten av konteringsarbeidet.

### Situasjon 3: Det nye forretningsområdet

Når foretaket trer inn i et nytt forretningsområde i tillegg til eller i stedet for gammel virksomhet, ser det

ut til at man ikke alltid greier å la regnskapssystemet endre seg tilsvarende. Eksempelene har vel flere sett: produksjonsbedriften som ble importør og distributør men som fremdeles har et regnskap som best modellerer et produksjonsforetak. Eller banken som har samme regnskapsmodell for eiendomsavdelingen som for den øvrige virksomhet. Hvorfor er det slik? Selvfølgelig kan årsaken være at den økonomiske ledelsen ikke er våken nok. Jeg tror imidlertid på en annen forklaring: Det regnskapssystem som bedriften benytter gir ikke nok fleksibilitet til at endringer i registreringsstruktur kan gjennomføres uten store kostnader. Noe av årsaken ligger i saldoorienteringen i regnskapet og noe i at bedriftens system tilbyr for få dimensjoner.

Der hvor bedriften har forretningsområder som er ulike, vil det som regel være slik at enkelte dataelementer bare har relevans for et av områdene. Det kan eksempelvis være slik at dataelementet «selger» ikke har mening i produksjonsavdeling A, bare i salgsavdelingen. Et godt bokføringssystem vil i registreringsammenheng og i rapportammenheng da utelate elementet selger i skjerm-bildet når brukeren

forts. side 24



tilhører produksjonsavdelingen. Vi sier at systemet kun viser brukeravhengige *utsnitt*. Skal dette være praktisk mulig, må bokføringssystemet være bygge ved hjelp av 4. generasjons programmeringsverktøy. Ved 3. generasjons programmeringsspråk som COBOL, PASCAL e.l. vil kostnadene ved å lage alternative registreringsbilder og rapporteringsmoduler lett bli for store.

En mer konvensjonell løsning på problemet ved ulike forretningsområder kan være å bruke ulike regnskapsløsninger som er spesielle for det enkelte området. Et område kan kjøre på et PC-system mens et annet område benytter et stort GL-system. Sammenhengen mellom områdene kan da beholdes ved en (konsern-) konsolidering. Dette kan være den rimeligste måten å ta hensyn til ulike informasjonsbehov på. Ulempen er at tilgangen til informasjon på detaljnivå bare finnes innen avdelingen, og at det er vanskelig å gjennomføre analyser som trenger data fra mer enn et forretningsområde. Foretakets styringsfilosofi avgjør om man kan tolerere denne ulempen.

#### Situasjon 4: Den nye eierstrukturen – Internasjonalisering

Mange har registrert at et foretak kan bli nødt til å avlevere årsregnskapet i henhold til flere standarder. Og at tolkningen av rapporten må foretas med varsomhet. Se f.eks. på følgende eksempel hentet fra The Accountant no 5815, July 1988:

#### Saatchi & Saatchi 1987 results

	UK GAAP USD M	US GAAP USD M
Net income	108.5	53.8
Shareholders equity	65.3	1020.2
Return on equity	166 %	5.2 %

Dette er selvfølgelig et ekstremt eksempel, og årsaken til forskjellen i dette regnskapet er forskjellig behandling av goodwill. Det er ikke mitt poeng i denne sammenheng at vi trenger kunnskap om alternative prinsipper. Dersom vi skal bli istand til å avlevere regnskapsrapporter i henhold til flere standarder *samtidig*, må transaksjonssystemene der regnskapsposterene fanges opp og lagres, gi den muligheten. Og dette betyr mer enn at vi kan ha norske og engelske navn på konti. En ting ser ut til å være nødvendig: Aggregeringsregler bygget på alternative standarder (ett rapporteringsprogram for hver standard). En ting ser ut til å være fordelaktig: dimensjoner (felt) i konteringen som gjør det mulig for rapporteringsprogrammene å velge korrekte behandlingsregler. Selv om det er mulig å konstruere de alternative rapporteringsprogrammene bare ved hjelp av en tradisjonell kontostreng, er konstruksjonen enklere og sikrere når vi har dimensjoner som tillater uavhengig kontering innen hver standard.

Dette øker kompleksiteten i registreringsarbeidet, og

## POSTKORT

Sendes ufrankert. Adressaten betaler porto
--

## SVARSENDING

Avtale nr. 133000/42

### PRAKTISK ØKONOMI

Bedriftsøkonomens Forlag A/S

Vaterland

0134 OSLO 1

vi kan igjen gjette på at et beslutningsstøttesystem for kontering kan være et rentabelt instrument.

Praksis idag er vanligvis at rapporteringen ved varierende standarder ikke håndteres av transaksjonssystemet, men av årsavslutningsprogrammer. Dette kan være en enkel og god løsning, men det er arbeidskrevende for høyt kvalifisert personell dersom man ønsker å foreta kortperiodisk eller løpende rapportering.

En annen strategi kan jo være å avvente harmonisering av internasjonale standarder, i første rekke mellom anglosaksiske og kontinentale standarder, men jeg er redd det kan bli lenge å vente.

### Syretesten

Jeg tenker ikke her på syretesten («the acid test») slik vi kjenner den i finansieringsanalysen. Det samme begrepet brukes (desverre) også om en metode for å bedømme nytten av regnskapsinformasjon. Det koster etterhvert meget å drive transaksjonssystemet, og når systemet blir stort øker sjansen for at deler av systemet bidrar lite til gode beslutninger i bedriften.

En kilde til ettertanke kan mange få ved å gjøre følgende: Velg tre viktige beslutninger fra siste års styreprotokoll. Forsøk å bedømme hvor stor andel av beslutningsgrunnlaget som var regnskapsinformasjon, og hvor stor del som stammet fra andre kilder. Gjett derefter på hva det kostet å få frem den informasjonen som ikke kom fra eget regnskap. Er det harmoni mellom ressursinnsats og verdi av informasjon? Er det slik at det er store informasjonsområder som regnskapet ikke dekker (men burde dekket)?

Syretesten har også en annen side: Forsøket på å eliminere rapporter og kontering som har begrenset verdi.

Min erfaring er at de fleste etter å ha gjennomført denne syretesten finner behov for å gjennomdrøfte sitt økonomiske styringssystem på ny. En slik drøfting kan føre til at man ønsker seg et flerdimensjonalt saldoløst system med gode muligheter for ad-hoc spørring.

### Forholdet mellom økonomisk analyse og transaksjonssystemet

Når nyutdannede økonomer kommer i jobb, får de ofte problemer med å anvende det metodeapparat for økonomisk analyse som var i pensum. Uten å invitere til debatt om den praktiske relevans av det som undervises, vil jeg nøye meg med å registrere en grunnleggende mangel ved regnskapssystemer slik de anvendes og tildels er konstruert.

Hva bruker vi mye tid på å undervise? Jo, forholdet mellom fysiske mengder og økonomiske målevariabler

vurdert i penger. Vi introduserer metoder og modeller for å avdekke samvariasjon mellom priser og mengder fordi dette har direkte virkning for bedriftens resultat. Hvorfor blir dette så vanskelig å bruke i praksis? Jeg mener årsaken er å finne i det forhold at *regnskapssystemene nesten ikke registrerer kvantum*. Altfor mange regnskapssystemer nøyer seg med skatteregnskapets krav til registrering av rene pengestrømmer. Årsaken til dette kan være at man lar kvantum behandles i forsystemene, og for å få så få transaksjoner som mulig fra forsystemet inn i bokføringssystemet aggregerer man slik at alle kvantumsdata fjernes. Hva må vi da falle tilbake på? Jo, komparative statiske analyser og tidsrekkeanalyser der fysiske kvanta sjelden inngår. Enkelt kan man kanskje si at vi kun opererer med *tiden* som forklarende variabel.

Bør det være slik? Hva jeg mener er vel åpenbart, men det er hva du mener som er avgjørende.

I de store norske bedriftene har vi akkurat vært gjennom en utskiftningsbølge av regnskapssystemer. De fleste har valgt transaksjonsorienterte systemer. Ikke sjelden har datateknologiens representanter vært pådriverne. For å foreta gode valg ved vurderingen av foreslåtte systemer må de som har ansvaret for økonomisk styring innse behovet for fleksibilitet i regnskapssystemet. Fleksibiliteten bør gi seg uttrykk i mulighet for endring i konteringsregler over tid, og mulighet for brukeravhengig utforming av skjermbilder og rapporter. □