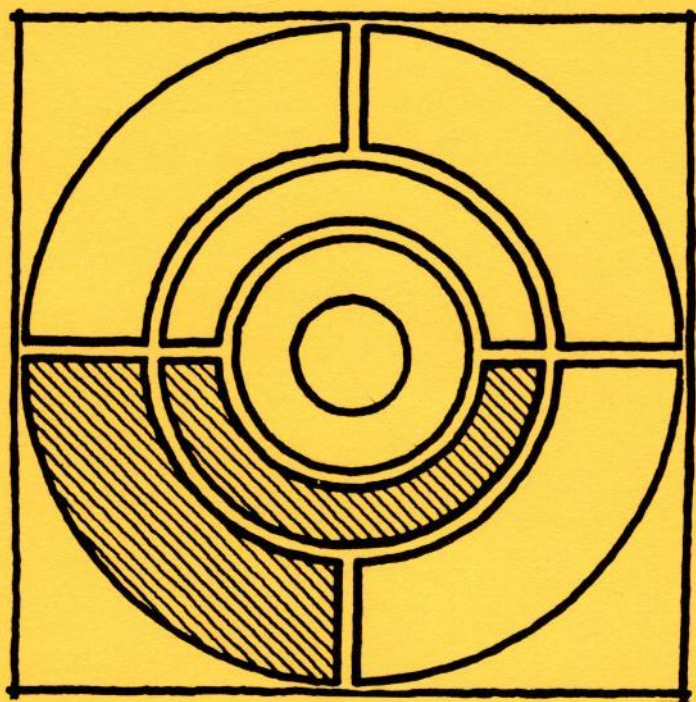




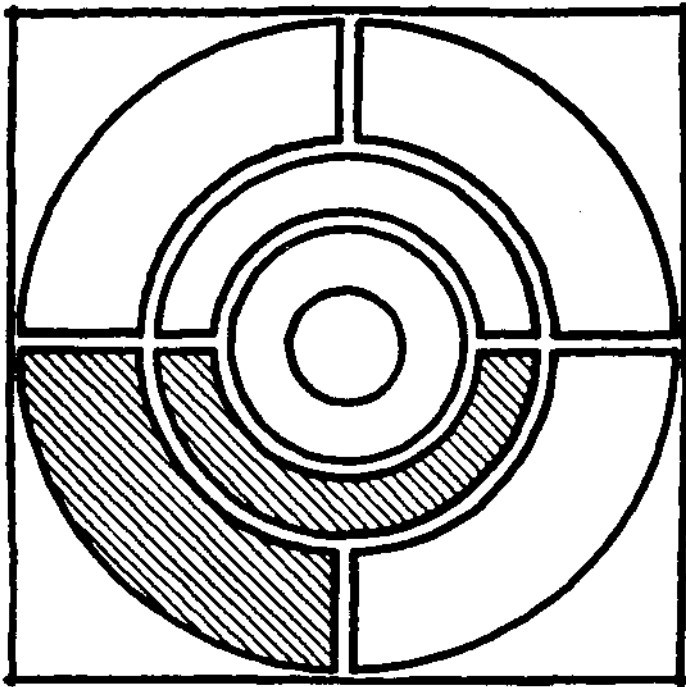
Dataanalyse



Dataanalyse



Dataanalyse



Dataanalyse

Dataanalyse

© Datacentralen A/S

Version 1, januar 1992

Ansvarlig:
Systemudviklingsservice

Fordelingsnøgle: **HDA1**

Indholdsfortegnelse

| | |
|---|----|
| Indledning og formål | 5 |
| Grundlæggende begreber og definitioner ... | 7 |
| Notation og konventioner for E-R-model ... | 12 |
| E-R-diagram | 12 |
| Beskrivelser | 19 |
| Tilstandsdiagrammer til livsforløb | 22 |
| Tabeller og normalisering | 23 |
| Fremgangsmåde for udarbejdelse af logisk datamodel | 29 |
| Direkte opstilling af E-R-model | 30 |
| Udledning ud fra funktionsmodel | 36 |
| Anvendelse af E-R-model i andre processer . | 38 |
| Databasemodel | 38 |
| Dataarkitektur og IRM | 40 |
| Matricer | 42 |

| | |
|------------------------------------|----|
| Mængder og frekvenser | 46 |
| Accesbeskrivelse | 48 |
| Stikord | 52 |

Indledning og formål

Denne brochures hovedsigte er at være en vejledning i anvendelse af E-R-modeller (entitets-relations-modeller). Der ud over beskriver den en række hjælpeteknikker, der er relevante i tilknytning til arbejdet med E-R-modeller. Det indbefatter emner, som kommer i forlængelse af, men ikke er direkte koblet til E-R-modellen, ved analyse af dataanvendelsen til designovervejelser.

Brochures formål er at beskrive metodens og teknikkernes indhold og anvendelsesområder.

Brochuren skal anvendes som vejledning og huskeliste, når systemudviklere med kendskab til metoden gennemfører analyse og design. Brochuren alene kan ikke give det kendskab til metoden, der er nødvendigt for at gennemføre dataanalyse med E-R-modeller. Dette må erhverves gennem uddannelse, f.eks. på Kursuscentrets kurser.

Brochuren er justeret efter de krav, der indgår i Datacentrals kvalitetsstyringssystemer i forbindelse med analyse og design.

Brochuren er opbygget, som følger:

- Først beskrives de grundlæggende begreber og definitioner, der knytter sig til E-R-modeller.
- Næste del beskriver notation og konventioner for E-R-modellen med E-R-diagram og tilhørende beskrivelser. Desuden beskrives tilstandsdiagrammer, som er en hjælpeteknik til livsforløb, og tabeller og normalisering, der er hjælpeteknikker til kontrol af modellen.
- Så beskrives fremgangsmåden for udarbejdelse af en logisk datamodel ved dataanalyse for en konkret udviklingsopgave. Der præsenteres den direkte opstilling med metoden hørende til E-R-modeller og en alternativ fremgangsmåde.
- Derefter beskrives, hvordan E-R-modeller kan anvendes i andre processer omkring udviklingen, dels i en tillempet form i det overordnede systemdesigns databasemodel, dels i generaliseret udgave i forbindelse med dataarkitektur og IRM.
- Endelig beskrives en række teknikker der bl.a. kan anvendes ved analyse af dataanvendelsen. Matrix-begrebet, beskrivelse af mængder og frekvenser samt accessbeskrivelser præsenteres.

Grundlæggende begreber og definitioner

Idegrundlag for E-R-model

Ideen i at anvende en E-R-model er, at man strukturerer data og sin forståelse af data i en model, der viser, hvad man har brug for at vide noget om. Den knytter dataene til de centrale begreber (entiteter), man har brug for at vide noget om. Da entiteterne er udtryk for den overordnede forretningside i den virksomhed, man analyserer, vil datamodellen være meget stabil, selv om anvendelsen af og behovet for data kan variere. De centrale entiteter ændrer sig kun hvis virksomheden skifter forretningsområde.

Med en E-R-model får man en model, der kombinerer grafik og tekst. De grafiske elementer giver overblik og sammenhæng. Teksten giver præcisering.

Elementer i en E-R-model

Entitet

“Noget”, et fysisk eller abstrakt begreb, der er af relevans for virksomheden, og som man har data om.

Relation

Viser sammenhænge mellem entiteterne. Disse sammenhænge er der, uafhængigt af hvordan dataene om entiteterne bruges.

Attribut

En egenskab ved en entitet eller relation. En attribut repræsenterer, hvad man mere præcist har brug for at vide om entiteten. De dataelementer, man har, indplaceres som attribut til en entitet eller relation.

Forekomst

Hvor en entitet repræsenterer det begreb, man har brug for at vide noget om, er en forekomst af entiteten de konkrete personer, ting, steder eller hændelser, man har data om. Forekomsterne opstår og forsvinder, medens selve entiteten hele tiden er der, og der vil normalt være flere forekomster af den.

Tilsvarende er en forekomst af en relation en konkret kobling mellem en forekomst fra hver af de relaterede entiteter.

Undertype

En underopdeling af en entitet i specialtilfælde. Entiteten er karakteriseret ved nogle attributter og/eller relationer, som er fælles for alle forekomster. Hver undertype har yderligere nogle attributter m.v., som er specielle for den.

Livsforløb

En *entitets livsforløb* omfatter dens fødsel, død og de andre hændelser, der påvirker den, samt de mellemliggende tilstande.

Fødsel: hvad er anledning til, at der opstår en ny forekomst af entiteten.

Død: hvad er anledning til, at den pågældende forekomst forsvinder igen.

Andre hændelser: ligger derimellem, og giver anledning til ændrede attributter og/eller relationer.

Tilstande: svarer til afhængigheder mellem hændelserne, hvor ikke alle sekvenser af hændelser er meningsfyldte. I en given tilstand er kun nogle bestemte hændelser mulige. Nogle hændelser giver skift i tilstand, andre gør det ikke.

Man kan også tale om *relationers livsforløb*. En relation kan etableres som følge af en hændelse, og kan på tilsvarende måde nedlægges igen.

Identifikation

En entitets identifikation er den måde, hvorpå man skelner de enkelte forekomster fra hinanden. Der kan gøres med en *identificerende attribut*, f.eks. nøglen i en tabel. Der kan også være *indirekte identifikation*, hvor en forekomst identificeres gennem den forekomst af en anden entitet (evt. de forekomster af et antal andre), den er relateret til, evt. i sammenhæng med en attribut.

Relationers værdi

Angiver hvor mange forekomster af en relation, der hører til én forekomst af en entitet, der indgår i relationen. For relationer mellem to entiteter findes typerne:

1:1 (en til en). For hver forekomst af entitet A er der højst en forekomst af entitet B, og omvendt.

1:m (en til mange). For hver forekomst af entitet A kan der være flere forekomster af entitet B. Den modsatte vej kan der højst være en.

m:m (mange til mange). For hver forekomst af entitet A kan der være flere forekomster af entitet B, og omvendt.

Man kan bruge en lignende notation til at vise grænseværdierne mere præcist, men da må man betragte hver retning af relationen separat:

0:1 angiver, at der til en kan være én, men ikke behøver være det.

1:1 angiver, at der skal være præcis én.

0:m angiver, at der kan være flere, men ikke behøver være nogen.

1:m angiver, at der skal være mindst én, men kan være flere.

Attributters værdi

De konkrete data til en givet forekomst af den entitet eller relation, attributten hører til. De fleste attributter har én (simpel) værdi for én givet forekomst.

En flerværdiet attribut er en attribut, som kan forekomme i et variabelt antal indenfor samme forekomst.

En *flerværdiet gruppe* er en sammenhørende gruppe attributter, der forekommer i variabelt antal. Det kaldes også for repeterende grupper.

Elementer i normalisering

Tabel

En struktur, der består af en gruppe søjler (også kaldet felter), og en variabel mængde rækker (forekomster). Til hver søjle er knyttet et domæne (=mulige værdier). En eller flere af søjlerne udgør en *nøgle* for tabellen, hvis der til én værdi af nøglen forekommer én og kun én række i tabellen.

Normalform

Regel, der lægger begrænsninger på den mulige opbygning af en tabel.

Der findes en række normalformer, hvor den i praksis vigtigste er 3. normalform. For at en tabel er på 3. normalform, skal en rækkes værdi for enhver søjle være entydigt bestemt ved:

1. værdien af tabellens nøgle,
2. hele en evt. sammensat nøgle,
3. kun denne nøgle (og ikke ved værdien af en anden søjle, som ikke er nøgle for tabellen).

Notation og konventioner for E-R-model

En E-R-model består af et E-R-diagram og beskrivelser af de elementer, der indgår i modellen.

E-R-diagram

Et E-R-diagram viser analyseområdets entiteter og relationerne mellem dem.

Her præsenteres to forskellige måder at opstille E-R-diagrammet på, dels med fuldt udbygget notation, dels i en simplificeret udgave. Endvidere vises arbejdspapirnotationen, der kun skal betragtes som hjælpemiddel under udviklingen.

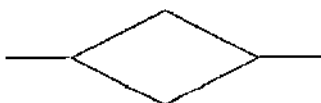
Fuldt udbygget E-R-diagram

Notation:

Entitet



Relation



Værdi på relation mellem to entiteter (med max. og min. for værdi)



0 eller 1



fra 0 til mange



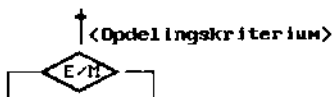
præcis 1



fra 1 til mange

Undertyper:

Entitet deles op



Markering i romben



viser sammenhængen undertypene imellem og til "overtypen":

E/- (exclusive), en forekomst af overtypen kan højst være i én af undertyperne.

I/- (inclusive), en forekomst af overtypen kan være i flere undertyper.

-/M(mandatory), en forekomst af overtypen skal være i mindst én af undertyperne.

-/O (optional), en forekomst af overtypen behøver ikke være en af undertyperne.

Attributter optræder ikke på E-R-diagrammet. Der kan være attributter knyttet til entiteter og til relationer.

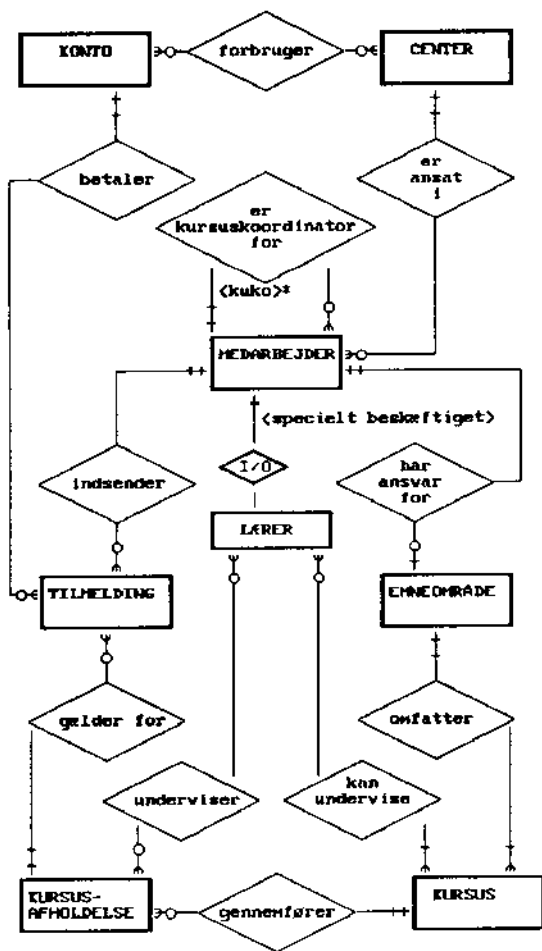
Navngivning:

Entiteter navngives med entalsformen af det begreb, de repræsenterer. De står for hvad man ved om en enkelt forekomst.

Relationer navngives, så de sammen med entiteternes navne fortæller, hvilken sammenhæng relationen er udtryk for. Det kan være vanskeligt at få forskellige navne til alle relationer. Det letter imidlertid referencer til den tilhørende beskrivelse. Entydige navne er nødvendige, hvis man vil bruge DesignAid's dictionary.

Attributter navngives efter, hvilken egenskab de beskriver. Det fulde navn refererer entiteten og beskriver egenskaben, f.eks. kursusafholdelse-start-dato. Det kan slutte med en data-klasse, som attributten tilhører. Så kan beskrivelsen af repræsentation foretages én gang for dataklasser som f.eks. datoer.

Eksempel:



* Angivelse af rolle på relation.
Kreves af DesignAid, når relation fra entitet til sig selv.

Simplificeret E-R-diagram

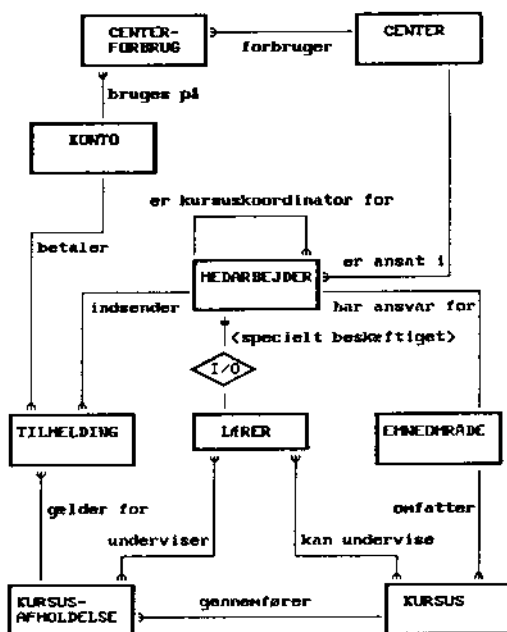
Et simplificeret E-R-diagram har en simplere notation:

- ingen rombe på relationer
- kun max. for værdi på relation

Relationer kan kun relatere to entiteter.

Der må ikke være attributter på relationer. Det er grunden til, at relationen "forbruger" i eksemplet nedenfor er blevet til entiteten "centerforbrug" og to nye relationer.

Eksempel:



Fordelene er, at det giver en simplere tegning at se på, og flere regler for om noget er en entitet eller relation.

En ulempe er, at det ikke kan valideres af DesignAid, og dermed ikke giver mulighed for "automatisk" balancering til funktionsmodellen fra struktureret analyse. Andre ulemper er, at modellen skal omorganiseres meget, hvis der senere kommer en attribut til en relation. Der bliver flere relationer at navngive, og der kan komme "tomme" entiteter.

Denne form er imidlertid velegnet til oversigt over systemets databaser i en databasemodel, når der anvendes et relationelt databasesystem. Det kan være et yderligere argument for at vælge den, hvis man vil videreudvikle den logiske datamodel til en databasemodel for systemet.

Arbejdsrapirnotation

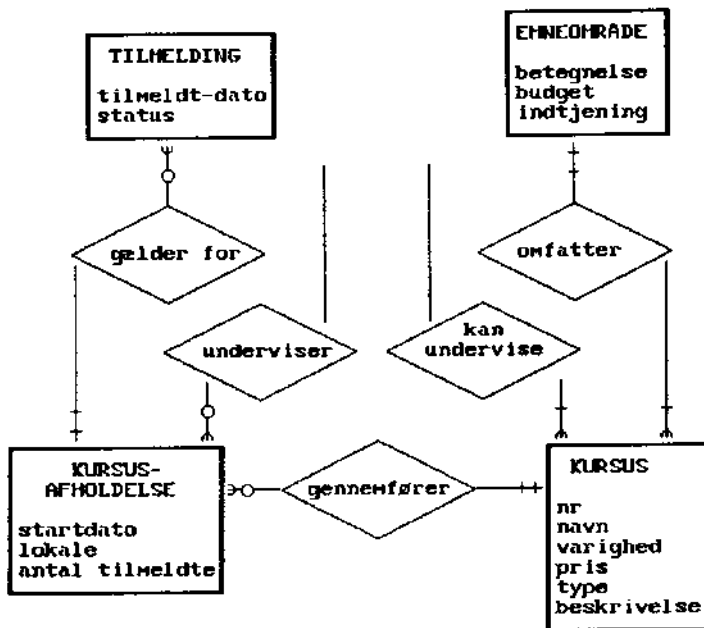
Dette hjælpemiddel er et E-R-diagram, hvor attributter også er skrevet på, ved siden af entiteten eller evt. inden i.

Under arbejdet med udformningen af modellen kan det være praktisk at have modellen eller dele deraf på denne form. Derved skal man ikke jonglere med så mange adskilte, ufærdige beskrivelseselementer. Det kan også være nemmere for involverede brugere at forholde sig til opdelingen, når de kan se, hvilke data der hører til de forskellige entiteter.

Men det fylder på papiret, og det bliver hurtigt uoverskueligt. Man får ikke et diagram, der giver overblik, til supplement af beskrivelsen af de enkelte elementer. Formen er uegnet som "leverance-dokumentation".

Eksempel:

(uddrag)



Beskrivelser

E-R-diagrammet skal suppleres med beskrivelser af de elementer, som indgår i modellen. Der skal være beskrivelser af entiteter, relationer og attributter. Beskrivelserne specificerer de enkelte elements væsentlige og relevante karakteristika.

Entiteter

Der skal være en oversigt over attributterne og identifikationen. Hvis man også anvender struktureret analyse kan listen skrives med notationen for dataordbogen. Man kan opfatte beskrivelsen som den del af dataordbogen, hvor registrene er beskrevet.

Under udviklingen bør man endvidere beskrive entitetens formål og rolle, med mindre systemet har et meget begrænset omfang. Denne beskrivelse skal vedligeholdes, hvis der er tale om et sammenhængende systemkompleks, hvor der gøres brug af fælles databaser.

Livsforløb skal overvejes og beskrives. Hvis funktionsanalysen udføres parallelt, og der holdes overblik med matrix eller lignende, behøver livsforløbet ikke beskrives i E-R-modellen. Hvis der er tale om et mere komplekst livsforløb med mange tilstande og afhængigheder, er det en fordel med en samlet beskrivelse.

Endvidere kan der være andre karakteristika, det er relevant at fastholde, f.eks. regler omkring godkendelse, yderligere

kriterier omkring oprettelse, ændring og nedlæggelse af forekomster.

Relationer

Relationernes væsentligste karakteristika er forholdet til entiteterne. Er der attributter, skal der være en liste over dem. Hvis navnet ikke er sigende nok, skal der være en uddybende beskrivelse af relationens formål.

På diagrammet vises værdi for relationer mellem to entiteter. En eventuel beskrivelse skal kun indeholde uddybende information, f.eks. mere specifikke værdier (Eksempel: "mange"=2), eller regler for deltagelse (Eksempel: Der skal altid være lærere, der underviser på de kursusafholdelser, der gennemføres).

Der kan endvidere være regler for hvilke forekomster, der kan relateres. Ellers må de være beskrevet i de funktioner, der opretter relationen. Koblingen mellem relationens livsforløb og entiteternes livsforløb kan beskrives nærmere.

For mange relationers vedkommende er oplysningerne trivielle. Man kan da vælge kun at beskrive nogle af relationerne yderligere, eventuelt i tilknytning til en entitetsbeskrivelse. Denne mulighed vil man normalt kun vælge, hvis man har besluttet at anvende det simplificerede E-R-diagram.

Attributter

Det skal beskrives hvilken egenskab, attributten repræsenterer, hvis navnet ikke er sigende nok. Endvidere kan man angive værdimængden for attributten.

En del regler kan med fordel placeres sammen med beskrivelsen af attributten, i stedet for ved funktioner, der tildeler attributten værdier. Det gælder, hvis reglerne er uafhængige af hvilken funktion, der foretager tildelingen.

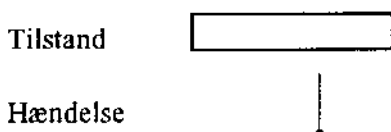
Uddybende bemærkninger til forståelse af attributtens placering og samspil med øvrige elementer kan tilføjes.

Attributbeskrivelsen vil være grundlag for databeskrivelser for implementerede systemer. Der tilføjes data-repræsentation i brugergrænseflade og internt.

Tilstandsdiagrammer til livsforløb

Hvis en entitets livsforløb omfatter flere tilstande, og der er afhængigheder mellem hændelserne, kan man vise samspillet ved hjælp af et tilstandsdiagram.

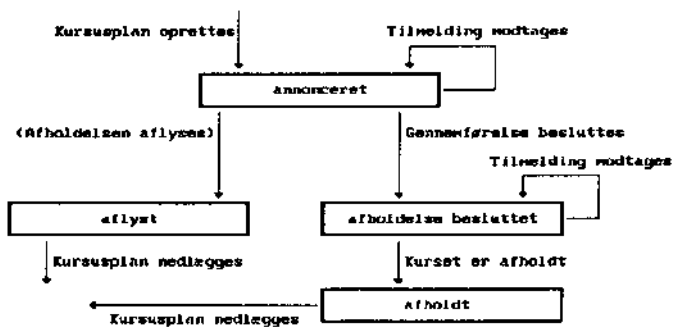
Notation:



På pilen noteres navnet på hændelsen. En pil, der ikke begynder eller slutter i en tilstand, angiver henholdsvis entitets fødsel og død. Bemærk, at ikke alle hændelser fører til en ny tilstand.

Eksempel:

Livsforløb for KURSUSAFHOLDELSE



Hvis der er mange hændelser, der ikke giver ændring i tilstand, kan de evt. skrives på en liste ved siden af diagrammet. Man kan vælge i stedet at lave en matrix over hændelser og tilstande, hvor det markeres hvilken tilstand, entiteten er i efter hændelsen.

De tilstandsdiagrammer, der benyttes til at beskrive livsforløb, har ingen forbindelse med dialognotationens tilstandsdiagrammer. Der er blot tale om diagramtyper, der ligner hinanden, men som bruges til vidt forskellige ting.

Tabeller og normalisering

Ud fra en E-R-model kan der dannes afledte tabeller, som kan anvendes til at eksemplificere forekomster (illustrationerne på de følgende sider viser afledte tabeller). De kan endvidere underkastes normalisering med henblik på kontrol af modellens korrekthed. Man behøver dog ikke nødvendigvis skrive tabellerne ud for at gennemføre normaliseringen.

Dannelse af tabeller

For en entitet dannes en tabel med attributterne som søjler.

For at finde en nøgle ser man på identifikationen af entiteten: Hvis der er en attribut eller en gruppe af attributter, som entydigt identificerer den enkelte forekomst, markeres den/disse

som nøgle. Hvis der er indirekte identifikation via relationer, trækkes nøglen fra den eller de relaterede entiteters tabel over i denne tabel, og markeres som nøgle sammen med eventuelle attributter, som indgår i identifikationen.

Hvis der er flerværdiede attributter eller grupper, skal de trækkes ud i en selvstændig tabel sammen med nøglen for den oprindelige tabel. Nøglen i den nye tabel bliver den oprindelige nøgle, udvidet med den flerværdiede attribut eller gruppes identifikation. Derved får alle tabeller fast længde, og er samtidig bragt på 1. normalform.

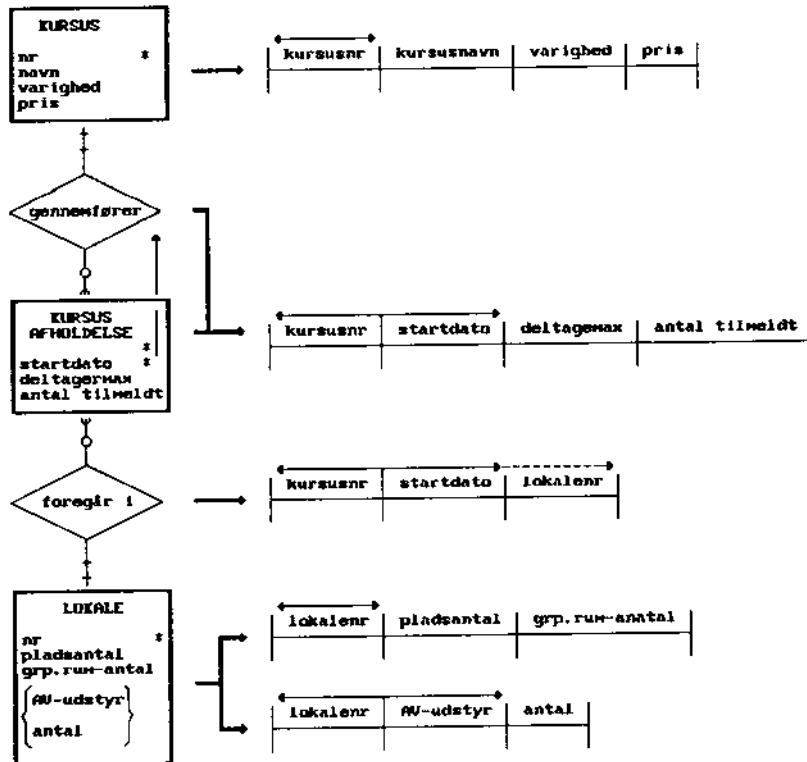
For at få relationerne med skal der skabes en kobling af nøglerne fra de relaterede entiteters tabeller. Dertil kan oprettes en selvstændig tabel med nøglerne fra entiteternes tabeller, hvor også relationens evt. attributter placeres som søjler.

Som nøgle for en relationstabel anvendes kombinationen af nøgler fra entiteternes tabeller. Ved en 1:m relation er dog egentlig "mange-endens" tabel-nøgle tilstrækkelig.

Eksempel:

(: = identifikation)

(← → = nøgle)



Normalisering til 3. normalform

Normaliseringsreglerne er illustreret ved nedenstående eksempel. De tre normalformer kan udtrykkes som betingelser, hvor registrene på hvert trin også skal opfylde betingelserne fra de foregående trin:

- 1.normalform: Alle felter skal være identificeret af nøglen.
- 2.normalform: Alle felter skal være identificeret af hele nøglen.
- 3.normalform: Ingen felter må være identificeret af andre felter end nøglen.

Den praktiske fremgangsmåde er at man gennemgår tabellerne og ser, om de opfylder 3. normalform. Hvis der er brud på reglerne, rettes tabellerne op, idet de først bringes på 1. normalform, derefter 2., osv. Når rettelserne er gennemført, indarbejdes deres konsekvenser i E-R-modellen.

Brud på normalformerne kan f.eks. være begrundet i:

- at en attribut er placeret på en forkert entitet eller relation,
- at den forkerte identifikation er udnævnt til nøgle,
- at man har overset en entitet eller relation,
- at man har blandet data hørende til forskellige entiteter.

Eksempel:

1. feltterne afhænger af NØGLEN

| FORKERT | ←→ | | sprog |
|---------|------|-----------------|---------------|
| | pnr | navn | |
| | 1914 | Søren Johannsen | engelsk, tysk |
| | 3117 | Juan Moreno | spansk |

| RIGTIGT | ←→ | | ←→ | |
|---------|------|-----------------|------|---------|
| | pnr | navn | pnr | sprog |
| | 1914 | Søren Johannsen | 1914 | engelsk |
| | 3117 | Juan Moreno | 1914 | tysk |
| | | | 3117 | spansk |

2. feltterne afhænger af HELE NØGLEN

| FORKERT | ←→ | | | | | |
|---------|----------|-----------|------------|--------|----------|-------|
| | kursusnr | startdato | kursusnavn | lokale | varighed | pris |
| | x4147 | 06-07-92 | Edb-intro | B. 14 | 2 dage | 3500 |
| | x4147 | 26-10-92 | Edb-intro | D. 12 | 2 dage | 3500 |
| | y2115 | 09-08-92 | Kvalitet | B. 17 | 5 dage | 11500 |

| RIGTIGT | ←→ | | | | ←→ | | |
|---------|----------|------------|----------|-------|----------|-----------|--------|
| | kursusnr | kursusnavn | varighed | pris | kursusnr | startdato | lokale |
| | x4147 | Edb-intro | 2 dage | 3500 | x4147 | 06-07-92 | B. 14 |
| | y2115 | Kvalitet | 5 dage | 11500 | x4147 | 26-10-92 | D. 12 |
| | | | | | y2115 | 09-08-92 | B. 17 |

3. feltterne afhænger af KUN NØGLEN

| FORKERT | ←→ | | | |
|---------|----------|------------|---------|----------------|
| | org.kode | betegnelse | chefpnr | navn |
| | 213312 | Salg | 1371 | Hans Jensen |
| | 313310 | Edb | 3987 | Peter Andersen |
| | 121410 | Marketing | 1371 | Hans Jensen |

| RIGTIGT | ←→ | | | ←→ | |
|---------|----------|------------|---------|------|----------------|
| | org.kode | betegnelse | chefpnr | pnr | navn |
| | 213312 | Salg | 1371 | 1371 | Hans Jensen |
| | 313310 | Edb | 3987 | 3987 | Peter Andersen |
| | 121410 | Marketing | 1371 | | |

Tabeller som "first cut design" til databasemodel

De afledte tabeller vil kunne bruges til "first cut design" af databaser. De afledte tabeller fra entiteterne, for m:m relationer og for relationer mellem mere end to entiteter, vil altid være et første bud på de databærende elementer i en database.

I en relationel database vil man typisk slå 1:m relationernes tabeller sammen med "mange-ende-entitetens" tabel, så denne tabel også rummer nøglen for den relaterede entitets tabel (en "fremmednøgle"). Tilsvarende gør man med eventuelle 1:1 relationer.

I en E-R-model, der benyttes til at vise en databasemodel, vil der være denne redundans. Relationerne vises på diagrammet, og fremgår samtidig gennem fremmednøgler i de tilhørende oversigter over dataelementer (attributter).

Fremgangsmåde for udarbejdelse af logisk datamodel

Den logiske datamodel, der udarbejdes ved dataanalysen i en udviklingsopgave, er det grundlæggende anvendelsesområde for en E-R-model.

Ved siden af dataanalysen skal de øvrige dele af analysen gennemføres. Til dette må der benyttes andre metoder - det indgår ikke i E-R-modellen.

I det følgende gennemgås, hvordan man udarbejder en E-R-model ved dataanalysen. Først beskrives den direkte opstilling af E-R-modellen i en selvstændig arbejdsproces, der kan foretages parallelt med de øvrige aktiviteter. Derefter vises en alternativ fremgangsmåde, hvor datamodellen udledes ud fra en funktionsmodel, der er udarbejdet først.

Direkte opstilling af E-R-model

Opstillingen af en E-R-model foregår i en iterativ proces, der fortsætter til man har et dækkende billede af analyseområdet. Man identificerer elementer til modellen, ser på deres karakteristika, finder yderligere elementer, og reorganiserer og reklassificerer allerede fundne.

Hovedtrinene ved opstillingen er følgende:

Find fundamentale entiteter

Hvilke centrale begreber har man brug for at vide noget om, f.eks. hvilke personer, ting, steder og hændelser? De kan ofte findes blandt:

- grundlæggende begreber, der er nødvendige for at forklare, hvad analyseområdet er for noget,
- interessenter,
- grundlæggende funktioners databehov.

Beskriv relationer mellem entiteterne

Relationer kobler entitetsforekomster, der har en sammenhæng. *Eksempel:* Medarbejdere og det center, de er ansat i.

Placer attributter

Hvilke attributter har man behov for, som mere konkrete oplysninger om entiteterne og deres samspil? Et dataelement indplaceres som attribut ét sted, ved den entitet eller relation, dataelementet beskriver.

Attributter kan evt. samles i logiske datagrupper, hvis der er en egenskab, som kræver et antal detailoplysninger.

Eksempel: Adresse- eller skatteoplysninger.

Nogle attributter kan være beregnbare ud fra de øvrige dele af modellen, f.eks. antallet af tilmeldinger til en bestemt kursusafholdelse. Hvis der imidlertid er tale om en egenskab, der hyppigt refereres til, kan man vælge at beholde attributten i modellen; men så må man markere, at det er en beregnbar attribut, og beskrive, hvordan den beregnes.

Undersøg identifikation af forekomster

Det skal være muligt at skelne de enkelte forekomster fra hinanden. Er der en identificerende attribut, eller en gruppe af attributter? Eller er der indirekte identifikation via en relation?

Ved manuelle rutiner kan det undertiden være vanskeligt at pege på identifikationen, da mennesker har nemt ved at arbejde med variabel, kontekst-afhængig identifikation. For eksempel kan en bestilling identificeres ved: Hvem der har bestilt, til hvornår, og - hvis det ikke er entydigt - også med hvad, der er bestilt.

Beskriv livsforløb

Hvad giver anledning til en entitets fødsel og død? Hvilke andre hændelser kommer entiteten ud for? Hvilke hændelser opretter og nedlægger f.eks. relationerne eller ændrer attributters værdi? Gennemløber en forekomst forskellige tilstande, og hvilke hændelser knytter de sig til?

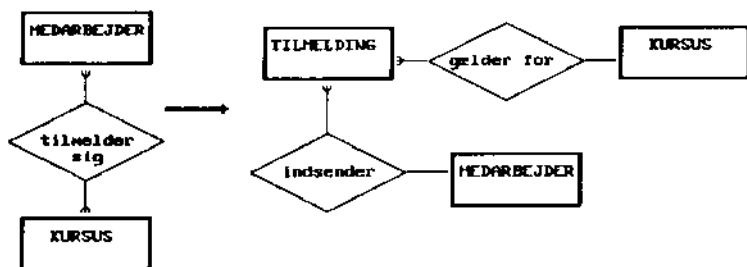
Identificer undertyper

Opdeling i undertyper giver mulighed for at beskrive en entitet som både et samlet begreb og forskellige varianter, som der knytter sig særlige attributter og relationer til. Med andre ord, at "der er noget, der kun gælder for nogle af dem".

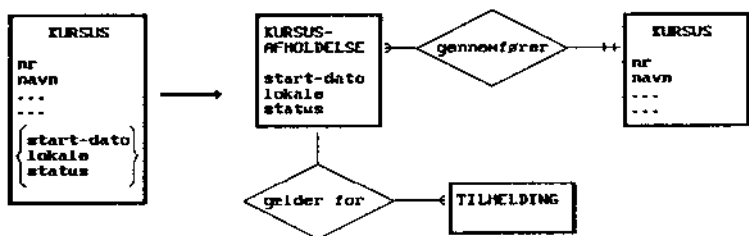
Undertyperne kan fremkomme ved at underopdele en entitet for at håndtere specialtilfælde. De kan også opstå ved at man samler beslægtede entiteter under en fælles hat. Man kan f.eks. samle ledere og projektdeltagere under det fælles begreb medarbejdere.

Videre bearbejdning og forfining af modellen

Man kan komme ud for, at en relation ved nærmere eftersyn viser sig at være et begreb som man omtaler selvstændigt, og evt. har yderligere relationer til. Relationen må da laves om til en entitet, der har relationer med de entiteter, den oprindelige relation koblede sammen. Eksempelvis tegnes først den relation, at en medarbejder tilmelder sig et kursus; men da tilmeldingerne modtages og behandles ved kursusadministrationen, må tilmelding altså være en entitet.

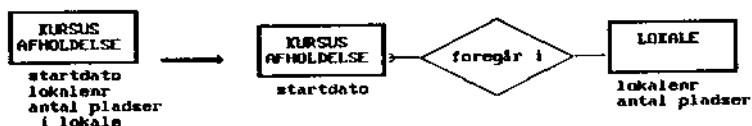


Grupperingen af attributter i en logisk datagruppe antyder i sig selv, at der er et interessant begreb. Der er måske tale om en entitet for sig? Specielt hvis det er en flerværdiet gruppe, og der er relationer, der går direkte på denne gruppe, så er det en entitet. Eksempelvis bliver en gruppe oplysninger om de enkelte afholdelser af et kursus til entiteten kursusafholdelse, da en tilmelding relaterer sig til en bestemt afholdelse.



En logisk datagruppe kan også være "data om data"; dvs. at dataene inden for en gruppe hører sammen med hinanden, men kun indirekte med den entitet, man først placerede dem i. Der var altså en entitet mere, som blot havde en relation

med den første. Eksempelvis oplysninger om lokalers kapacitet og indretning i forhold til kursusafholdelser, der foregår i et lokale.



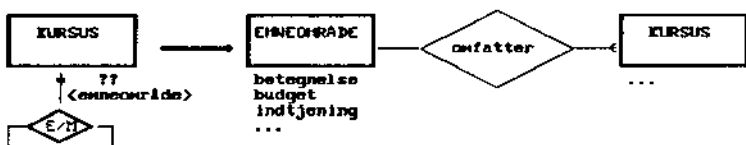
Vedrørende identifikation af entiteter skal man, specielt i ældre edb-systemer, være opmærksom på fænomenet "data-bærende nøgler". Det er en nøgle, hvor værdien af nøglen - eller dele af nøglen - har en selvstændig betydning, udover at identificere forekomsten. Eksempelvis repræsenterer et cpr-nummer mindst tre attributter: en entydig identifikation af personen, fødselsdato og køn. Attributterne skal udskilles og beskrives hver for sig. Databærende nøgler kan give store og kostbare problemer i et system, både ved brug og vedligeholdelse.

Normalt er der flere forekomster til en entitet. En undtagelse er den såkaldte scope- eller system-entitet, der repræsenterer analyseområdet som sådant. Til denne entitet kan være knyttet globale attributter. Det kunne f.eks. være virksomhedens navn og adresse, eller sådan noget som momsprocent eller sats for feriegodtgørelse. Entiteten beskrives, men behøver ikke tegnes med på E-R-diagrammet.

At formulere hændelserne i et livsforløb kan identificere yderligere attributter, som det er nødvendigt for entiteten at kende. Hvis et livsforløb omfatter forskellige tilstande, vil der være attributter, der fortæller om tilstanden.

Man kan blive opmærksom på, at der kan opstå andre entiteter i tilknytning til et livsforløb. Eksempelvis at oplysninger om en forekomst ved entitetens død overføres til, eller indgår i, dannelse af forekomster af andre entiteter, der tjener historik- eller statistik-formål.

Overvejelserne omkring undertyper kan give anledning til at identificere en type som en selvstændig entitet. I kursusseksemplet findes begrebet: emneområde for kurser. Det er en planlægnings- og opfølgingsenhed, som der derfor er knyttet data til.



Samspil med andre processer i analysen

Det skal sikres, at funktionsanalysens analyseområde ikke går ud over området for dataanalysen. Selv om datamodellen omfatter hele datagrundlaget for funktionsmodellen, kan funktionsanalysen give anledning til, at der bliver identificeret nye dataelementer eller entiteter.

Analyse af hændelser indgår også i funktionsanalysen, så på dette område kan arbejdet på de to modeller supplere hinanden direkte. Overvejelserne om livsforløb er f.eks. med til

at sikre, at alle nødvendige funktioner huskes, bl.a. funktionerne til håndtering af fødsel og død.

Hændelserne kan under analysen klassificeres som normale eller unormale. Derved kan man overbevise sig om, at normalforløbet er dækket. Dette kan også bruges i diskussion af nødvendige funktioner; hvor meget skal edb-systemet f.eks. skelne og håndtere de unormale funktioner.

Hvis analyseområdet deles op i et antal delområder, kan man opstille en model for hver af disse, der viser det pågældende delområdes syn på data. Disse forskellige syn integreres i en samlet model for hele analyseområdet.

I den samlede model beskrives entiteter, relationer og attributter. De forskellige syn viser, hvilke af dem der er relevante for et delområde. Jævnfør også samspillet mellem dataarkitektur og IRM's emne-niveau i forhold til projektmodeller.

Udledning ud fra funktionsmodel

Man kan vælge at opstille en E-R-model på grundlag af en detaljeret funktionsmodel. Man tager da udgangspunkt i de dataelementer, der er i de foreløbige registre. På denne måde bruges dataanalysen alene til at placere de allerede identificerede, nødvendige dataelementer i logiske registre.

Fremgangsmåden er som følger:

1. Der opstilles en skitse med de relevante entiteter og relationerne mellem dem. Dataelementerne grupperes som hørende til entiteterne. De listes, evt. i tabelform, og der markeres nøgler.
2. Derefter gennemføres en normalisering. De ændringer til tabellerne, dette giver anledning til, indarbejdes som nye og/eller ændrede entiteter og relationer med dataelementerne som attributter.
3. Der tegnes et E-R-diagram med de identificerede entiteter og relationer. Beskrivelserne omfatter listen over attributter og markering af nøglen som identifikation.

Ved denne fremgangsmåde får man ikke det fulde udbytte af E-R-modellens muligheder til at strukturere sin forståelse af data. Man får en model, der knytter sig tæt til de kendte funktioner. Man udnytter heller ikke, at det skiftende fokus mellem arbejdet på funktionsmodel og arbejdet på datamodel giver gensidig inspiration og kontrol.

Fremgangsmåden er relevant, hvis der er skal udvikles et system, der er beregnet til at være lukket, og der er foretaget en grundig analyse af funktionerne i forbindelse med forberedelsen af opgaven. En model, der er udarbejdet på denne måde ud fra en funktionsmodel for et system, kan indgå som udgangspunkt for opbygning af en mere omfattende model.

Anvendelse af E-R-model i andre processer

E-R-modeller kan også anvendes i andre processer omkring udviklingen af systemer, hvis man fortolker lidt om på betydningen af de elementer, der indgår. I det følgende beskrives sådanne anvendelser, dels som databasemodel, dels i forbindelse med dataarkitektur og IRM.

Databasemodel

Til det overordnede systemdesigns databasemodel kan man anvende en tillempet form for E-R-model.

Databasemodellen svarer stort set til den logiske datamodel. Den praktiske fremgangsmåde vil derfor være at kopiere og tilrette den logiske datamodel for at udlede databasemodellen, jvf. også muligheden for afledte tabeller som "first cut design".

En entitet i den tilrettede model er en veldefineret gruppe data, der lagres (et register, individ, tabel, record, segment eller hvad det nu hedder i det aktuelle system). Attributterne er de enkelte datafelter i gruppen. Man får så et E-R-diagram der viser, hvilke af den slags entiteter der findes. Beskrivelsen af de enkelte entiteter lister dataelementerne og har evt. anden information, der kan overtages fra den logiske model. Til datafelterne knyttes den valgte repræsentation og evt. information om den tilsvarende attribut.

Relationerne repræsenterer en umiddelbar forbindelse mellem to entiteter, hvor man kan fremfinde de sammenhørende forekomster. Ofte vil relationerne ikke have navne internt i systemet. De navngives heller ikke på diagrammet, medmindre de umiddelbart overtager et navn fra den logiske model. I de fleste databasestyresystemer er det kun muligt at have 1:m relationer.

Ud over de data, der repræsenterer den logiske datamodel, kan der være data, som knytter sig til den valgte implementering af systemet. Disse hjælperegistre skal også indgå i databasemodellen, og tilføjes som entiteter.

Til visse typer databasestyresystemer er der tradition for at benytte en anden notation for relationer end i den simplificerede form for E-R-diagram. Til netværks-databaser benyttes pile, som i et accesdiagram. Til hierarkiske databaser bruges en konvention, hvor den relative placering på papiret angiver hierarkiet. Der henvises i øvrigt til materiale om de konkrete produkter.

Dataarkitektur og IRM

E-R-modeller er udviklet med henblik på dataanalyse for en konkret udviklingsopgave, men de har vist sig - i en let generaliseret udgave - at være velegnede til de overordnede modeller, der benyttes i dataarkitektur og Information Resource Management (IRM).

Dataarkitektur og IRM opfatter data og information (behandlede data) som en fælles ressource for hele virksomheden, som skal beskrives, struktureres og styres.

Man har brug for en samlet beskrivelse for at have overblik over de ønskede og tilgængelige ressourcer. Én model for en hel virksomhed på det detaljeringsniveau, der benyttes i dataanalysen, er umulig at overskue og holde ajour. I stedet kan man anvende en E-R-model med en mere overordnet opfattelse af entitetsbegrebet. En entitet er her et samlet område for information. Attributterne er de typer af information, der indgår i området. Relationer mellem disse overordnede entiteter viser, at der er en direkte relation mellem de underliggende entiteter.

Sådanne modeller kan udarbejdes i flere niveauer. De typer af information, som attributterne repræsenterer, udgør hver et samlet område for information, som igen kan beskrives mere specifikt. På det underliggende niveau vises relationerne mellem de forskellige typer af information, og hvilke af dem, der har relationer til andre entiteter.

Det niveau, hvor entiteterne svarer til de logiske datamodeller, med konkrete attributter og forekomster med værdier, kaldes også emne-niveauet.

Med baggrund i den samlede IRM-models emne-niveau kan man definere projekt- eller systemmodeller, ved at plukke de entiteter, som er relevante for det pågældende projekt eller system. Vedrørende definitionen af entiteterne refereres til emnemodellerne.

Grænsen mellem de to begreber dataarkitektur og IRM er ikke helt veldefineret. Betegnelsen dataarkitektur benyttes, når man koncentrerer sig om de "formaliserbare" data, man benytter i sine systemer. IRM har egentlig et bredere sigte. Man taler i princippet om alle former for information og behandling af den. Vedrørende mere specifikke retningslinier og vejledninger, henvises til speciallitteratur.

Matricer

Matrix-begrebet kommer fra området dataarkitektur og IRM. Matricer er imidlertid en generelt anvendelig teknik, der kan bruges i mange sammenhænge. En væsentlig anvendelse er som data/funktionsmatrix, der er velegnet til overblik over data-anvendelsen og samspillet mellem E-R-model og funktionsmodellen fra struktureret analyse.

Notation og konventioner

Matricerne bygger på krydsreference-princippet. Der er imidlertid mere information end blot et kryds i felterne.

Principskitse:

| Emneområde B | Emneområde A | | | | | |
|--------------|--------------|----|----|-----------------------------|----|-----|
| | A1 | A2 | A3 | A4 | A5 | ... |
| B1 | | | | | | |
| B2 | | | | | | |
| B3 | | | | information om sammenhængen | | |
| B4 | | | | | | |
| ⋮ | | | | | | |

Data/funktionsmatrix

I en data/funktionsmatrix er emnerne entiteter og funktioner. I feltet vises, hvad funktionen gør med entiteten: Opret, Læs, Ret og/eller Slet.

Det vises med bogstaverne CRUD (for Create, Read, Update, Delete) eller med mere og mindre udfyldte boller (for opret, ændring eller læs): ● ○ resp. ⊗ ○

Eksempel:

| Entitet ↑ Funktion ↓ | Kursus | Kursusafholdelse | Lærer | Tilmelding | ... | ... |
|-------------------------|--------|------------------|-------|------------|-----|-----|
| Opret kursusplan | ○ | ● | | | | |
| Registrer tilmelding | ○ | | | ● | | |
| Endelig bekræftelse | ○ | ⊗ | ⊗ | ○ | | |
| Aflys kursus | ○ | ⊗ | | ⊗ | | |
| ... | | | | | | |
| ... | | | | | | |

Anvendelse igennem et udviklingsforløb

Ved opstilling af de første skitser til logiske modeller kan en matrix være en god hjælp. Som supplement til en hændelsesliste, der i forbindelse med struktureret analyse benyttes til at identificere logiske funktioner, kan den give ideer til nye entiteter og hændelser.

Undervejs i et udviklingsforløb, hvor der arbejdes parallelt med en datamodel og en funktionsmodel, er opstilling af en matrix en hurtig måde at kontrollere, om modellerne stadig hænger sammen.

I systembeskrivelsen kan en data/funktionsmatrix benyttes til at vise koblingen mellem hovedfunktioner og entiteter. Den giver et hurtigt overblik, og kan med fordel anvendes i systemer, hvor funktionsmodellerne ikke er ført helt op på det overordnede niveau.

Andre typer matricer kan også benyttes. Det kan f.eks. være matricer for bruger/data eller bruger/funktion i forbindelse med autorisation og sikkerhed.

Anvendelse ved dataarkitektur og IRM

En af de væsentligste anvendelser for matricer er til at identificere ansvarsfordeling for data. Det er her data/funktionsmatricen kommer ind.

En ansvarsplacering er en nødvendig forudsætning for at kunne udnytte dataene som fælles ressource. Ansvar for en overordnet entitet placeres samme sted som ansvar for den funktion, der opretter dens underliggende forekomster.

Under udarbejdelsen af de overordnede modeller kan matrixerne bruges som middel til at identificere overordnede entiteter.

Matricer anvendes også til at vise sammenhængen mellem de overordnede, logiske modeller, de eksisterende systemer, planlagte systemer og virksomhedens eller institutionens organisation.

Man kan f.eks. lave matricer for:

- entitet/register
- entitet/system
- register/system
- logisk forretningsfunktion/system
- organisatorisk enhed/logisk forretningsfunktion
- organisatorisk enhed/system

Mængder og frekvenser

Opgørelse af datamængder og frekvenser for brugen af dem er et led i dataanvendelsesanalysen. Det er udgangspunktet for dimensionering og optimering ved design af systemet.

Datamængder

Den enkleste form for opgørelse af datamængder er at angive antallet af forekomster for hver entitet. Det suppleres med den forventede udvikling, angivet som vækst eller fald. Man kan også specificere fordelingen på til- og afgang, hvis bestanden er meget skiftende. Dette detaljeringsniveau vil normalt være tilstrækkeligt.

Ovennævnte optællingsmetode omfatter imidlertid ikke relationerne, og disse kan også være betydningsfulde. Ved valg af implementeringsmåde kan det være af betydning, hvor stor en del af forekomsterne, der har en bestemt relation.

Man kan desuden, for en given entitet, beskrive den statistiske fordeling af forekomsterne af dens relaterede entiteter og eventuelle flerværdiede attributter og grupper. Denne information kan bl.a. bruges til at optimere databasedesignet, ved at afsætte plads til et bestemt antal forekomster.

Brugen af data

Mængder og frekvenser opgøres normalt i forhold til funktionerne. For hver logisk funktion ser man på hvor hyppigt den udløsende hændelse forekommer, og hvordan forekomsterne fordeler sig over tiden. En helt detaljeret opgørelse på brugen af data kræver, at man sammenkæder funktionsoplysninger og datamængder.

Man kan få et mere sammenhængende billede af onlinefunktionerne i et edb-system ved at tilføje oplysninger om antallet af samtidige brugere, samt brugernes normale anvendelse af de enkelte funktioner, det såkaldte funktionsmix. Fordelingen over tid kan specificeres i større eller mindre grad. Det almindeligste er en separat opgørelse for spidstime(r), samt en gennemsnitsværdi for den øvrige arbejdstid.

Batch-funktioner angives med ind- og uddatamængder samt kørselsfrekvens.

Accesbeskrivelse

Accesbeskrivelser er detaljerede angivelser af funktioners tilgang til data. De benyttes til kontrol af E-R-modellen og i forbindelse med designovervejelser. De er normalt ikke en del af den blivende dokumentation.

Accesdiagram:

Til accesbeskrivelse anvendes accesdiagrammer. På et accesdiagram vises accesobjekterne, accesvejene mellem dem og indgangsvejen(e). Dette kan suppleres med en specifikation af, hvilke data der anvendes fra accesobjekterne.

Notation:

Accesobjekt



Accesvej



(der findes én)

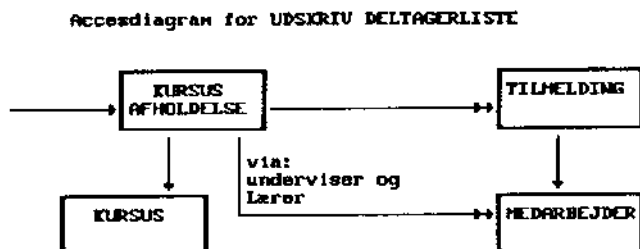


(der findes flere)

Accesobjektet er et databærende element, hvorfra der skal bruges data. I en E-R-model er det de entiteter og relationer som indeholder de attributter, der skal bruges.

Accesvej er den vej, man kommer til accesobjekterne.
I en E-R-model går accesvejene via relationerne.

Eksempel:



Anvendelse til kontrol af E-R-model

Hvis man har mistanke om, at en funktions databehov ikke kan dækkes med den foreliggende model, kan man be- eller afkræfte dette ved at tegne et accesdiagram for funktionen.

Man kan eventuelt tegne "ovenpå" et E-R-diagram, i stedet for at udarbejde et selvstændigt diagram.

Detaljerede analyser i forbindelse med designovervejelser

Accesdiagrammer kan benyttes til detaljerede analyser af dataanvendelsen. Disse analyser foretages normalt på data-

basemodellen eller et udkast til denne. Analysen koncentrerer sig om de dele af systemet, hvor designeren har mistanke om, at der kan opstå problemer.

Hvis man kombinerer accesserne med datamængder, og samler beskrivelser for flere funktioner, bliver det muligt at identificere særligt belastede accesveje mv.

Det vil sjældent være relevant at præsentere disse detaljerede beskrivelser for kunden. Det er interne arbejds papirer for specialister, og kan evt. indgå i den tekniske dokumentation henvendt til systemudviklere og systemforvaltere, for at vise forudsætningerne for det valgte design.

Stikordsregister

Symboler

- 0:1 10
- 1:1 10
- 1:m 10, 39
- 3. normalform 11, 26

A

- Accesdiagram 48
- Accesobjekt 48
- Accesvej 48
- Arbejdspapirnotation 17
- Attribut 8, 10, 21, 30
- Attributters værdi 10

B

- Beregnbar attribut 31

D

- Data/funktionsmatrix 43
- Dataarkitektur 40, 44
- Databasemodel 28, 38
- Databærende nøgle 34
- Datamængder 46
- Død 9

E

- Entitet 7, 19, 30

F

Flerværdiet attribut 10, 24
Flerværdiet gruppe 11, 24, 33
Forekomst 8, 23, 34
Forskellige syn 36
Fremmednøgle 28
Fuldt udbygget E-R-diagram 13
Funktioner 35, 36, 47
Fødsel 9

H

Hændelse 9, 22, 35

I

Identificerende attribut 9
Identifikation 9, 31, 34
Indirekte identifikation 9
IRM 40, 44

L

Livsforløb 9, 19, 22, 31, 34
Logisk datagrube 31, 33

M

m:m 10

N

Normalisering 11, 26
Nøgle 11, 23

R

Register 39, 45

Relation 8, 10, 20, 30, 32
Relationers værdi 10

S

Scope-entitet 34
Simplificeret E-R-diagram 16
System-entitet 34

T

Tabel 11, 23, 28
Tilstand 9, 22

U

Undertype 8, 13, 32, 35

DataCentralen