DIKU rapport 92/10

Fuldtekstsøgesystemer til fagfolk
- med fokus på faciliteter der støtter systemernes udvikling over tid

Morten Hertzum og **Henrik Søes**

Datalogisk Institut ved Københavns Universitet, DIKU 1992

Resumé

Dette speciale handler om fuldtekstsøgesystemer, der skal fungere som redskaber for fagfolk. Fuldtekstsøgning betegner søgning direkte på de ord, der forekommer i søgesystemets dokumenter, dvs søgning i dokumenternes fulde tekst. Med fokuseringen på fagfolk lægges særlig vægt på fleksibilitet og gennemskuelighed, samt på at søgesystemet integreres i brugerens samlede arbejdssituation.

Vi fokuserer primært på ét basalt forhold i arbejdssituationen: Den ændrer sig over tid. Vi ser et stort behov for faciliteter, der giver mulighed for, at disse ændringer indarbejdes i søgesystemet, så snart brugeren bliver opmærksom på dem. Det er efter vores mening en nødvendighed, hvis det skal lykkes at integrere søgesystemet i brugerens arbejdssituation. Vi behandler tre faciliteter, der giver denne mulighed: Et dynamisk emneregister, der giver brugeren mulighed for løbende at foretage ændringer i grupperingen af dokumenterne i søgesystemet; en dynamisk tesaurus, der giver mulighed for løbende at tilpasse systemet til ændringer i sprogbrug og begrebsdefinitioner; og egne notater, der løbende kan indføjes i den allerede eksisterende dokumentsamling.

Med vores fokusering på udvikling over tid stiller vi særlige krav til de værktøjer, søge-systemet baseres på. De skal give mulighed for, at udvidelser og ændringer så vidt muligt er enkle at foretage og kan gennemføres uden at give anledning til en række følgeændringer andre steder i søgesystemet. Vi har en idé om, at den ønskede funktionalitet og fleksibilitet kan opnåes ved at basere fuldtekstsøgesystemer på relationsdatabaser. Denne mulighed er af ret ny dato og stort set upåagtet i kommerciel sammenhæng. I litteraturen betragtes relations-databaser generelt som underlegne i sammenligning med de systemer, der er udviklet specielt med henblik på lagring af og søgning i tekst. Vi analyserer og vurderer, hvorvidt det er frugtbart at basere fuldtekstsøgesystemer på relationsdatabaser.

Specialet består for det første af en beskrivelse af *state of the art* for fuldtekstsøge-systemer. Her er det de tekniske aspekter, vi lægger vægt på. Den anden del af specialet er en case indenfor området juridisk informationssøgning. Vi indleder denne case med en indkredsning af juridisk sagsbehandling - den brugssituation juridisk informationssøgning indgår i - og de krav, ønsker og behov jurister har i forbindelse med juridiske informations-søgesystemer. Derefter designer og implementerer vi en prototype på et juridisk fuldtekst-søgesystem. Datagrundlaget for prototypen er stillet til rådighed af Karnovs Forlag og omfatter 400 sider (ca 4 Mb tekst) fra Virksomheds-Karnov.

Vi mener, det har vist sig lovende at gøre udvikling over tid til en nøglefaktor i designet af et fuldtekstsøgesystem til fagfolk. Vi giver en række eksempler på, hvordan de udviklede faciliteter til støtte af udvikling over tid giver brugeren mulighed for at bringe søgesystemet i tættere kontakt og større overensstemmelse med sin arbejdssituation. Vi mener desuden, at relationsdatabaser har vist sig lovende som grundlag for søgesystemer, der på en fleksibel måde skal kunne udvikle sig med brugssituationen. Denne mulighed opnåes samtidig med tilfredsstillende svartider og et nok stort, men fuldt ud acceptabelt pladsforbrug.

Forord

Dette projekt er lavet på Datalogisk Institut ved Københavns Universitet, DIKU, i perioden september 1990 - juni 1991. Projektet er vores speciale.

Specialet handler om fuldtekstsøgesystemer, der skal fungere som redskaber for fagfolk. Hermed lægges særlig vægt på menneske/maskine-samspillet og på, at søgesystemet integreres i brugerens samlede arbejdssituation. I den sammenhæng ser vi specielt et behov for faciliteter, der tillader søgesystemerne at udvikle sig over tid. Vi vil behandle tre sådanne faciliteter: Brugeren skal have mulighed for løbende at foretage ændringer i grupperingen af søgesystemets dokumenter, for løbende at tilpasse systemet til ændringer i sprogbrug og begrebsdefinitioner, og for løbende at tilføje nye tekster.

Vi har en ide om, at den ønskede funktionalitet og fleksibilitet kan opnåes ved at basere fuldtekstsøgesystemer på relationsdatabaser. Denne mulighed er af ret ny dato og skyldes dels de stærkt faldende priser på store mængder hurtig lagerplads, dels at effektive relations-databasesystemer nu er tilgængelige. I specialet vil vi - ved hjælp af relevant litteratur og implementering af en prototype - analysere og vurdere mulighederne for at udvikle fuld-tekstsøgesystemer, der støtter udvikling over tid. Specielt vil vi vurdere og analysere realismen i at basere sådanne søgesystemer på relationsdatabaser.

Specialet består af et litteraturstudie, der fører til en beskrivelse af *state of the art* for fuldtekstsøgesystemer, og af en case. Emnet for vores case er juridisk informationssøgning i forbindelse med den juridiske sagsbehandling. Denne case munder ud i udviklingen af en prototype på et juridisk fuldtekstsøgesystem, kaldet Edb-Karnov. Den afgørende grund til, at vi vælger juridisk informationssøgning som emne for vores case, er, at Karnovs Forlag har givet os mulighed for at basere vores case på et fantastisk relevant datamateriale. Dette datamateriale - godt 400 sider fra Virksomheds-Karnov - har givet os mulighed for at implementere og afprøve Edb-Karnov på et realistisk grundlag. Vi er direktør Jens Peter Nielsen, Karnovs Forlag, megen tak skyldig for denne mulighed.

Vi skylder også jurist Dorthe la Cour, Dansk Arbejdsgiverforening, og advokatfuldmægtig Per Sjøqvist, advokatfirmaet Horten & Co, tak for deres imødekommenhed og engagement i forbindelse med vores interviews om indholdet af juridisk sagsbehandling og juristers krav og ønsker til juridiske informationssøgesystemer. Desuden skal Jens Peter Nielsen, Per Sjøqvist og forsker Peter Ingwersen, Biblioteksskolen, have tak for engageret deltagelse i vores demonstrationer af Edb-Karnov. Sidst, men ikke mindst, tak til Erik Frøkjær for inspirerende vejledning. Det var ham, der gav os ideen til specialet og etablerede kontakten til Karnovs Forlag.

Morten Hertzum og *Henrik Søes*

Indholdsfortegnelse

1. Indledning og problemformulering

Dette projekt handler om fuldtekstsøgning. Fuldtekstsøgning betegner søgning direkte på de ord, der forekommer i søgesystemets dokumenter, dvs søgning i dokumenternes fulde tekst. Fuldtekstsøgning betegnes ofte også fritekstsøgning. Vi foretrækker termen fuldtekst-søgning, da vi mener fritekstsøgning antyder nogle muligheder og frihedsgrader, søgesystemerne ikke kan leve op til. Vi opfatter snarere fritekstsøgning som en salgsbetegnelse. Fuld-tekstsøgning sættes ofte i modsætning til nøgleordsbaseret søgning, hvor hvert dokument beskrives ved en gruppe nøgleord, og søgningerne udelukkende omfatter disse nøgleord. Den gruppe nøgleord, der knyttes til et dokument, udgør en fortolkning af dokumentets indhold. Denne fortolkning er præget af, hvad dokumentet hidtil har været brugt til, og dækker derfor kun sjældent alt, hvad det senere vil blive brugt til. En væsentlig fordel ved fuldtekstsøgning er, at der ikke ligger en sådan fortolkning mellem brugeren og dokumenterne - søgningerne foregår direkte i dokumenternes tekst. Da fuldtekstsøgning giver brugeren direkte adgang til dokumenternes tekst, er der endvidere mulighed for ved skimming af de fremfundne dokumenter straks at sortere mange af de irrelevante fra.

Nøgleordsbaseret søgning var veletableret længe før edb blev et hjælpemiddel i informationssøgningen. Fuldtekstsøgning er derimod en mulighed, der er opstået med brugen af edb. Fuldtekstsøgesystemernes udbredelse vokser endvidere i takt med, at flere og flere tekster udarbejdes ved hjælp af tekstbehandlingssystemer og derfor ikke skal genindskrives, når de skal lægges ind i et fuldtekstsøgesystem. Fuldtekstsøgning er således let og billigt at etablere. Det skal specielt vurderes i forhold til nøgleordsbaseret søgning, hvor bestemmelsen af nøgleordene er en ressourcekrævende opgave, der som oftest må varetages af personer med stor faglig indsigt.

Et centralt problem ved fuldtekstsøgning er, at brugerne er afhængige af, hvilke ord forfatterne af systemets dokumenter har brugt. Ordvalget er meget person- og kontekst-afhængigt; men ordvalg, begrebsdefinitioner, terminologi osv ændrer sig også med tiden. Det gør det svært at vælge søgeord til forespørgslerne. Vi vil senere se på, hvordan dette problem kan reduceres ved hjælp af en avanceret søgefacilitet: En tesaurus. Et eksempel på en tesaurus er *Computing Reviews Classification System*, den systematik som ACM klassificerer edb-litteraturen efter. En tesaurus er en begrebsordbog, men i modsætning til en normal ordbog beskrives begreberne ved deres relationer til andre begreber, ikke ved definitioner. Fire typiske relationer i en tesaurus er bredere, snævrere, synonyme og relaterede begreber. Formålet med en tesaurus er primært at generere flere søgeord ud fra dem, brugeren angiver; herved øges chancen for sammenfald mellem søgeordene og ordvalget i dokumenterne.

Ovenfor er såvel vigtige fordele som alvorlige ulemper ved fuldtekstsøgning trukket frem. Gennem sammenligninger med nøgleordsbaseret søgning vil vi flere gange i løbet af specialet søge at vurdere fordele og ulemper ved fuldtekstsøgning. I dette kapitel præsenterer vi den sammenhæng, de begge indgår i: Området informationssøgning. I kapitlets andet afsnit gør vi rede for det perspektiv, vi betragter informationssøgning i. Det er som redskaber for fagfolk, vi er interesserede i søgesystemer. Derefter præciserer vi formålet med specialet gennem en problemformulering. De to centrale ideer er faciliteter, der støtter søge-systemets udvikling over tid, og brugen af relationsdatabaser som grundlag for sådanne systemer. Kapitlet afsluttes med en oversigt over projektets opbygning og indhold.

1.1 Informationssøgning

Informationssøgning kan foretages helt uden elektroniske hjælpemidler; i dette projekt

bruges ordet imidlertid udelukkende i betydningen edb-baseret informationssøgning. Informationssøgning er et omfangsrigt område; det rækker fra tekniske problemstillinger omkring udformningen af søgealgoritmer over kognitive undersøgelser af menneskers søgeadfærd til sociale og organisatoriske spørgsmål om de forhold, søgesystemerne anvendes under. I dette projekt er det de tekniske problemstillinger, der er i centrum. De andre aspekter inddrages også; men det sker primært som led i en indledende bestemmelse af, hvilke ønsker og behov et søgesystem bør opfylde og tage hensyn til. I dette afsnit behandler vi imidlertid informationssøgning generelt, dog rettes opmærksomheden i løbet af afsnittet mod fuld-tekstsøgning. Vi starter med at se på den proces, der fører frem til formuleringen af en forespørgsel.

Data, information og utilstrækkelig viden

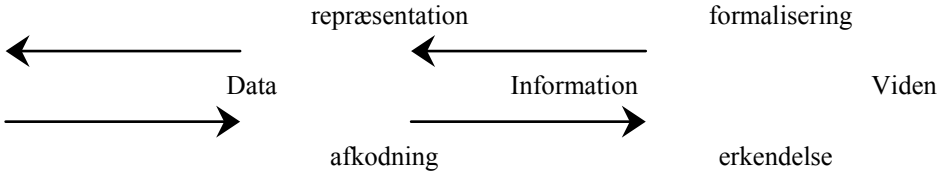
Hvad får mennesker til at stille forespørgsler til søgesystemer? En væsentlig forudsætning for, at det sker, er naturligvis en vis tro på, at søgesystemet kan levere de ønskede oplysninger. Mere fundamentalt udspringer forespørgslerne af, at man erkender en tilstand med util-strækkelig viden. Belkin (1980) kalder denne tilstand *the anomalous state of knowledge* (ofte forkortet ASK). Belkins beskrivelse af ASK ligger meget tæt op af Mackay, der allerede i 1960 beskrev denne tilstand af utilstrækkelig viden (Mackay 1960):

> *[Tilstanden kan for en persons vedkommende beskrives som] a certain incompleteness in his picture of the world, an inadequacy in what we might call his "state of readiness" to interact purposefully with the world around him.*

En forespørgsel til et søgesystem udspringer således af utilstrækkelig viden; søgesystemet indeholder imidlertid ikke viden, men data. Bindeleddet mellem viden og data er information: Erkendelsen af den utilstrækkelige viden konkretiseres og formuleres i et informationsbehov, og søgesystemets data fortolkes - ofte helt uden problemer - som meningsfuld information. Data, information og viden er fundamentalt forskellige størrelser. Forskellene træder imidlertid ofte i baggrunden til fordel for retorik og store visioner om, hvad edb-systemerne i en nær fremtid vil kunne. Det er særlig udbredt indenfor forskningen i kunstig intelligens, jævnfør fx begrebet 'vidensbase' og hele forestilingen om at udvikle 'intelligente systemer'. Da forskningen i kunstig intelligens siden midten af 80'erne har vundet udbredelse indenfor informationssøgning (Ingwersen 1987), finder vi det værd at uddybe forskellen mellem data, information og viden.

Der findes talrige definitioner af de tre begreber, *"Dictionary of Computing"* (1990) og *"Edb-ordbog"* (1971) indeholder fx gode og anerkendte definitioner af data og information. Vi har mødt og diskuteret en række sådanne definitioner, siden vi begyndte på DIKU. Det følgende bygger på disse definitioner og diskussioner og på (Lindgreen 1988).

Data er konkrete fysiske fænomener, fx dokumenterne i et fuldtekstsøgesystem. Viden er et abstrakt fænomen, der kan opdeles i formaliserbar og ikke-formaliserbar viden. Den ikke-formaliserbare viden kaldes ofte tavs viden. Viden opnåes gennem erkendelse - lagring af oplysninger er ikke tilstrækkeligt - og kan derfor alene besiddes af mennesker. Information er et abstrakt fænomen og i modsætning til data og viden en ustabil tilstand. Information eksisterer kun under kommunikation som en overgangsform mellem viden og data:



Figur 1.1. Data, information og viden. En del af en persons samlede viden kan formaliseres og derved blive til kommunikerbar information. Information kan repræsenteres, fx på tryk, og derved blive til data, der kan lagres, transmitteres osv. Data kan afkodes af dem, der kender koden, derved får dataene mening og bliver til information. Ex: Tekst kan afkodes af dem, der kan læse. Endelig kan information erkendes og derved blive en del af en persons viden.

Før en bruger kan rette en forespørgsel til et søgesystem, må den utilstrækkelige viden formaliseres, dvs udtrykkes som et informationsbehov, og repræsenteres, dvs formuleres som en forespørgsel i søgesystemets forespørgselssprog. Det kræver som regel en række overvejelser og reformuleringer, før forespørgslen svarer blot nogenlunde til den utilstrækkelige viden, den udspringer af. To brugere, der oplever den samme utilstrækkelige viden, vil sjældent stille identiske forespørgsler. Den formalisering og repræsentation, der fører frem til forespørgslen, er således en kompliceret og subjektiv proces. Også brugerens tilegnelse af de data, søgesystemet giver som svar på en forespørgsel, er en subjektiv proces. Både afkodning og erkendelse afhænger af personen; der er tale om fortolkning, ikke om personuafhængig overførsel af information/viden.

Det følger af det ovenstående, at vurderingen af hvilke dokumenter, der er relevante i forbindelse med en given forespørgsel, kun kan foretages af brugeren: Relevansen skal vurderes i forhold til brugerens utilstrækkelige viden, ikke i forhold til den stillede forespørgsel. Relevans er således også en subjektiv størrelse. Specielt har søgesystemets fremfinding af dokumenter intet med en relevansvurdering at gøre. Der kan højst være tale om at forsøge at efterligne menneskers relevansvurderinger ved hjælp af avancerede algoritmer. Søgesystemet finder alene dokumenter frem ud fra sammenligninger af data.

Anvendelsesområder for fuldtekstsøgning

Fuldtekstsøgning anvendes indenfor en lang række områder. Anvendelserne omfatter blandt andet:

Juridiske informationssøgesystemer. Juridiske informationssøgesystemer har længe været det førende område indenfor fuldtekstsøgesystemer (Tenopir 1984). De to mest omtalte systemer er LEXIS (fra 1973) og WESTLAW (hvor fuldtekst blev tilføjet i 1978). På det seneste er forskningen blevet rettet mere og mere mod AI-teknikkerne: Systemerne skal være ekspertsystemer og 'ræsonnere' ud fra loven og en beskrivelse af sagen, se fx (Cross & DeBessonet 1985).

Bibliografiske søgesystemer. Biblioteksverdenen præges af klassifikation, fx efter decimalklassifikationssystemet, og indeksering med nøgleord. Men på grund af den stadigt voksende mængde litteratur, der skal læses og klassificeres/indekseres, er der også interesse for fuldtekstsøgesystemer. Her slipper de, der udbyder søgesystemet, for arbejdet med at afgøre, hvad teksten handler om (Eddison & Batty 1988).

Arkiveringssystemer. Integrationen af fuldtekstsøgesystemer i kontorautomation er et af de væsentlige, nyere anvendelsesområder for fuldtekstsøgesystemer (Faloutsos 1985). Fuldtekstsøgesystemer kan bidrage med muligheder for effektiv opbevaring og fremfinding af store mængder tekst. Udfordringen er imidlertid primært integrationen; søgesystemerne skal sammenbygges med systemer, der tjener andre formål og har en anden funktionalitet.

Elektroniske aviser og tidsskrifter. Her er fuldtekstsøgesystemerne opstået i takt med,

at aviser og tidsskrifter har indført tekstbehandling og lignende i journalisters og typografers arbejde. Fuldtekst benyttes til fordel for indeksering, dels fordi indeksering er en ekstra udgift, dels fordi indeksering tager ekstra tid, og det anses for afgørende, at de elektroniske aviser altid er ajourført (Tenopir 1984).

Elektroniske ordbøger/leksika (Marchionini 1989). Der er to hovedbegrundelser for at benytte elektroniske ordbøger og leksika fremfor dem på bogform. Den ene er de langt bedre muligheder for at holde oplysningerne ajour. Den anden er de bedre søgemuligheder. Det bliver fx muligt at søge i forklaringen på et opslagsord fremfor blot efter forklaringen på et opslagsord.

Programbiblioteker. Genbrug af kode forudsætter velorganiserede programbiblioteker. Søgning efter programmer/programstumper med en given funktionalitet er typisk og kan støttes ved søgning i kommentarerne til programmerne (Maarek & Smadja 1989). Søgning efter implementeringsdetaljer kræver imidlertid andre faciliteter.

1.2 Redskaber til fagfolk

Det er som redskaber for fagfolk, vi er interesserede i informationssøgesystemer. Fagfolk adskiller sig såvel fra personer uden faglig kompetence som fra personer med mere rutineprægede opgaver. Fagfolk har stor frihed i udøvelsen af deres daglige arbejde. De er ikke underlagt en stram styring og kontrol; deres indsats bestemmes først og fremmest af deres personlige engagement og ansvarlighed. Fagfolks arbejdsopgaver er komplicerede og kan ikke formaliseres - det er den primære årsag til fagfolkenes frihed i tilrettelæggelsen og udførelsen af deres arbejde. Fagfolks arbejde kan således ikke reduceres til anvendelse af regler; en væsentlig del af deres kompetence er intuition og sund fornuft opnået gennem erfaring (Dreyfus & Dreyfus 1986).

Indenfor forskningen i kunstig intelligens er det en udbredt opfattelse, at også fagfolks kvalifikationer kan udtrykkes i regler og formalismer. Det er denne opfattelse, der ligger til grund for fx ekspertsystemerne. AI-forskerne har imidlertid endnu ikke kunnet fremvise noget resultat, der kan sandsynliggøre denne opfattelse. De fleste ekspertsystemer er prototyper, der ikke bruges i praksis, og ingen af de eksisterende ekspertsystemer besidder eksperternes kvalifikationer. De bedste ekspertsystemer kan være en god støtte for eksperterne; men erstatte eksperterne er der ingen af dem, der kan. Der er endvidere blevet argumenteret overbevisende for den principielle umulighed i at formalisere fagfolks og eksperters kvalifikationer, se fx (Dreyfus & Dreyfus 1986), (Winograd & Flores 1986) og specielt for det juridiske område (Leith 1986).

Efter vores mening er der ingen tvivl om, at fagfolk ikke kan erstattes af edb-systemer. Edb-systemerne skal derimod støtte brugerne i deres arbejdsproces; de skal fungere som redskaber for brugerne. Det er ud fra denne synsvinkel, vi betragter søgesystemer. Det betyder, at brugeren skal have fuld kontrol over søgningerne og søgesystemets øvrige funktioner. Grænserne for systemets funktioner fastlægges således af, at intet må foregå bag om ryggen på brugeren. Hvis brugeren skal have fuld kontrol over søgesystemet, er det afgørende, at det er klart for brugeren, hvad systemet kan, hvordan det gør det, og hvordan man får det til at gøre det. Det stiller store krav til brugergrænsefladen og til, at systemets funktioner er velvalgte og umiddelbart forståelige for brugerne. Udformningen af systemets funktioner styres således af, at funktionaliteten skal være fuldstændig gennemskuelig for brugeren.

Jurister og juridisk sagsbehandling er et godt eksempel på fagfolk og deres arbejde. Frøkjær & Pedersen (1987) understreger dette ved at trække tre punkter frem: Det første er nødvendigheden af ræsonnementer baseret på sund fornuft. Det andet er manglen på en

egentlig model for juridiske afgørelser. Det tredje punkt er kompleksiteten af jura og lovgiv-ning; denne kompleksitet skyldes blandt andet, at lovteksterne er fyldt med vagt definerede begreber. Juridisk sagsbehandling præges endvidere af, at der er meget store mængder tekst, der muligvis indeholder relevante informationer. Alene love, bekendtgørelser og cirkulærer udgør et omfattende tekstmateriale; men derudover er juristen også afhængig af betænkninger og andre forarbejder til lovteksterne, af principielle domsafgørelser, af nedskrevne kutymer osv. Et godt informationssøgesystem vil således kunne yde juristen værdifuld støtte. Vi mener derfor, at juridisk informationssøgning er et velegnet område for specialets case.

1.3 Problemformulering

Dette speciale handler om fuldtekstsøgesystemer, der skal fungere som redskaber for fagfolk. Vi har to centrale ideer om sådanne søgesystemers funktionalitet og opbygning: Vi ser et stort behov for faciliteter, der tillader søgesystemerne at udvikle sig over tid, og vi ser en mulighed for, at den ønskede funktionalitet og fleksibilitet kan opnåes ved at basere sådanne søgesystemer på relationsdatabaser. Gennem specialet vil vi analysere og udvikle disse to ideer samt vurdere, hvorvidt de er frugtbare og realistiske.

Vi bruger begrebet *udvikling over tid* til at betegne det basale forhold, at den kontekst, et søgesystem indgår i, ændrer sig med tiden. Tre eksempler kan illustrere dette: Brugernes syn på og grupperinger af søgesystemets dokumenter ændres, fordi det nu er andre arbejdsopgaver, der giver anledning til søgningerne. Sprogbrug og begrebsdefinitioner ændres, så tesaurussen, hvis der er sådan en, bliver utidssvarende, og det bliver sværere at vælge de rigtige søgeord, når søgningerne også omfatter ældre dokumenter. Nogle af dokumenterne i søgesystemet bliver forældede, og der opstår nye vigtige dokumenter, som også burde ligge i systemet og være søgbare. Et godt søgesystem må støtte udvikling over tid, så ændringer i konteksten kan afspejles i søgesystemet efterhånden, som brugeren bliver opmærksom på dem.

Vi vil behandle tre faciliteter, der tillader søgesystemer at udvikle sig over tid: Et dynamisk emneregister, der giver brugeren mulighed for løbende at foretage ændringer i og tilføjelser til den systematik, der - med inspiration fra en bogs indholdsfortegnelse - giver en oversigt over søgesystemets dokumenter. En dynamisk tesaurus, der giver brugeren mulighed for løbende at tilpasse søgesystemet til ændringer i sprogbrug og begrebsdefinitioner. Egne notater, der giver brugeren mulighed for løbende at føje nye tekster til søgesystemet.

Relationsdatabaser blev introduceret af Codd i 1970 (Codd 1970) til håndtering af strukturerede data. Det varede imidlertid omkring 15 år, før de blev implementeret effektivt og begyndte at slå igennem kommercielt. I den mellemliggende periode blev de afvist med den begrundelse, at de var for ressourcekrævende til at være praktisk anvendelige. Nu har relationsdatabaser etableret sig som standarden for lagring og organisering af strukturerede data. I forbindelse med informationssøgesystemer afvises de imidlertid stadig med henvisning til, at de koster for meget tid og lagerplads. Vi mener, relationsdatabasernes funktionalitet og fleksibilitet gør dem velegnede som grundlag for søgesystemer, der skal kunne udvikle sig over tid. Vi afviser ikke kritikken, men mener som udgangspunkt, at fordelene mere end opvejer ulemperne.

Specialet omfatter for det første et litteraturstudie, der skal munde ud i en beskrivelse af *state of the art* for fuldtekstsøgesystemer. For det andet vil vi diskutere og analysere *state of the art* i forhold til vores ideer om udvikling over tid og brugen af relationsdatabaser. Det vil vi gøre gennem en case, der omfatter design, konstruktion og implementering af en prototype på et juridisk fuldtekstsøgesystem - Edb-Karnov. Edb-

Karnov er en edb-udgave af Karnovs Lovsamling og bliver afprøvet på et datamateriale, der udgør godt 400 af lovsamlingens ca 4000 tætskrevne sider.

Sammenfattende vil vi analysere og evaluere problemer med og ideer til udvikling af fuldtekstsøgesystemer med fokus på faciliteter, der tillader systemerne at udvikle sig over tid.

1.4 Oversigt over specialets opbygning og indhold

I næste kapitel - kapitel 2 - vil vi gøre rede for *state of the art* for fuldtekstsøgesystemer. Vi indleder med en kort oversigt over de hovedstrømninger, der har været i forskningen og udviklingen af søgesystemer de sidste 30-40 år. Derefter beskrives de væsentligste elementer i fuldtekstsøgesystemer; det drejer sig om lagringsteknikker, tekstrepræsentation, søgeteknikker og faciliteter til at forbedre søgningerne. Vi kommer også kort ind på søgesystemernes brugergrænseflade. Beskrivelsen af *state of the art* udgør den tekniske del af grundlaget for den følgende case.

I kapitel 3 behandles juridisk sagsbehandling og juristers krav til juridiske informationssøgesystemer. I dette kapitel, der er det første i behandlingen af vores case, forsøger vi at indkredse den arbejdssituation, vores case udspiller sig i. Som et led i denne indkredsning beskriver vi to af de hjælpemidler, jurister har til rådighed - Karnovs Lovsamling og Retsinformation. Hen igennem kapitlet lokaliseres og formuleres en række krav og forventninger til juridiske søgesystemer. Kapitlet munder ud i en første beskrivelse af vores prototype på et juridisk fuldtekstsøgesystem, Edb-Karnov.

Emnet for kapitel 4 er design og konstruktion af Edb-Karnov. Formålet med Edb-Karnov er at analysere og vurdere, hvorvidt vores ideer, om faciliteter der støtter udvikling over tid, kan gøres frugtbare og realistiske. Edb-Karnov er således baseret på en relationsdatabase og omfatter tre faciliteter, der muliggør udvikling sig over tid: Et dynamisk emneregister, en dynamisk tesaurus og egne notater. Kapitlet afsluttes med en afprøvning af Edb-Karnovs svartider på 30 testforespørgsler af varierende kompleksitet.

I kapitel 5 vender vi os mod ajourføringen - udgivelsen af nye udgaver af Edb-Karnov. Brugeren vil ikke benytte mulighederne for at foretage løbende ændringer i fx tesaurussen, hvis disse ændringer går tabt, når der installeres en ajourført udgave af tesaurussen. Det, at brugeren løbende kan ændre i tesaurussen, har således kun værdi, hvis det integreres med udgivelsen af nye udgaver. Vi kommer vi med et forslag til, hvordan denne integration kan finde sted.

I kapitel 6 afsluttes vores case med, at Edb-Karnov demonstreres for såvel læseren som en potentiel udbyder, en potentiel bruger og en forsker indenfor informationssøgning. Denne demonstration har for det første til formål at give et overblik over prototypen og dens funktioner. For det andet udgør demonstrationen en evaluering af Edb-Karnov. Denne evaluering består af de indbudte personers kommentarer til og kritik af systemet.

I kapitel 7 gør vi rede for de metodiske overvejelser, der ligger bag specialet. Endelig er kapitel 8 sammenfatningen på specialet.

Efter kapitel 8 følger en annoteret litteraturliste. Specialet indeholder også en række bilag. Bilag 1 er en ordliste, hvor en lang række af specialets centrale begreber er forklaret i kort form. Bilag 2 indeholder den stopliste, vi har lavet og brugt i forbindelse med Edb-Karnov. Bilag 3 og 4 er et resultat af de to interview, vi afholdt for at indkredse juridisk sagsbehandling og juristers krav til juridiske informationssøgesystemer. De to bilag indeholder henholdsvis interviewguiden og de to interviewreferater. Bilag 5 indeholder en oversigt over de testforespørgsler, der indgik i afprøvningen af Edb-Karnovs svartider. Endelig indeholder bilag 6 en oversigt over Edb-Karnovs pladsforbrug.

2. *State of the art* for fuldtekstsøgesystemer

Emnet for dette speciale er fuldtekstsøgesystemer til fagfolk. Fagfolk stiller efter vores mening nogle specielle krav; men de grundliggende elementer i fuldtekstsøgning er de samme, hvad enten der er tale om fagfolk, eller brugergruppen er en anden. Dette kapitel skal danne baggrund for, at vi i de efterfølgende kapitler selv designer og konstruerer en prototype på et fuldtekstsøgesystem. Kapitlet dækker derfor såvel fuldtekstsøgning generelt som aspekter, der mere eller mindre er specielle for fuldtekstsøgesystemer til fagfolk. Hovedvægten ligger på de aspekter, der er særlig relevante for specialets hovedproblemstillinger. Vi lægger således særlig vægt på diskussionen af relationsdatabasers anvendelighed som grundlag for fuldtekstsøgesystemer og diskussionerne af tesaurusser og andre faciliteter, der kan støtte søgesystemernes udvikling over tid. Det er endvidere de tekniske problemstillinger, vi behandler i dette kapitel. Fuldtekstsøgning omfatter også andre problemstillinger, fx kognitive og organisatoriske; de berøres ikke eller kun indirekte.

Ordet 'fuldtekstsøgning' betegner søgning direkte på de ord, der forekommer i sammenhængende naturlig tekst. Fuldtekstsøgning sættes ofte i modsætning til nøgleordsbaseret søgning, som er baseret på de nøgleord, der er hæftet på teksten. Indenfor denne modsætning bruges ordet 'fuldtekstsøgning' imidlertid i flere varierende betydninger. Betydningerne varierer med hensyn til, hvor stor en del af teksten der indgår i søgningerne. Nogen opfatter fuldtekstsøgning som søgning i dokumenternes fulde tekst. Her står fuldtekstsøgning i modsætning til såvel nøgleordsbaseret søgning som søgning, der kun omfatter en del af dokumenternes tekst. Andre bruger også fuldtekstsøgning som betegnelse for søgning, der kun omfatter en del af den fulde tekst, fx resuméet. Her står fuldtekstsøgning kun i modsætning til nøgleordsbaseret søgning, og en mere velvalgt betegnelse ville efter vores mening være 'tekstsøgning'.

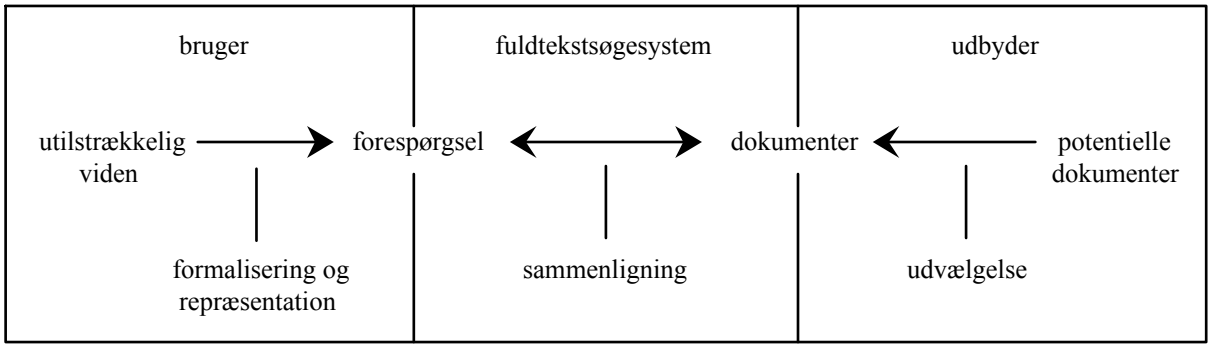
Betydningerne varierer endvidere med hensyn til, hvorvidt eventuelle figurer, billeder og lignende er med. I mange fuldtekstsøgesystemer svarer et dokument til en artikel, der også udgives på papirform. Dokumentet og artiklen er imidlertid sjældent identiske, da figurer, billeder osv typisk ikke er med i dokumentet. Adskillelsen af på den ene side tekst og på den anden side figurer, billeder og lignende er kunstig og opstået, fordi den gør implementeringen af søgesystemerne lettere; der er imidlertid både behov og muligheder for at integrere de to (Ashford 1986). Til trods for det vil vi ikke forsøge en sådan integration; vi afgrænser os fra at behandle figurer, billeder og lignende yderligere.

Endelig må der skelnes mellem på den ene side systemer, hvor hele teksten findes i systemet, og på den anden side systemer, hvor søgningen foregår i hele teksten (Tenopir 1984). De to er ikke nødvendigvis ens: Et søgesystem kan være baseret på, at læsningen foregår on-line og derfor indeholde hele teksten; søgning kan imidlertid udelukkende ske ved nøgleord. Vi vil benytte følgende definition af fuldtekstsøgesystemer:

Fuldtekstsøgesystemer er den type informationssøgesystemer, hvor dokumenternes fulde tekst (eksklusiv eventuelle figurer, billeder og lignende) er til rådighed i systemet, og søgning foregår i hele teksten.

Fuldtekstsøgesystemer indgår i en kommunikationsproces mellem på den ene side dokumenternes forfattere og de, der er ansvarlige for søgesystemet, og på den anden side brugerne. Vi opstiller nedenstående figur som en referenceramme for resten af kapitlet; figuren giver et overblik over de væsentligste elementer i den situation,

fuldtekstsøgesystemer anvendes i:



Figur 2.1 Situationen ved anvendelse af fuldtekstsøgesystemer. De to aktører er: Brugeren som skal udtrykke sin utilstrækkelige viden i en forespørgsel, og udbyderen som skal vælge de dokumenter, systemet skal indeholde. I midten ses fuldtekstsøgesystemet, som skal finde dokumenter frem ud fra en sammenligning af forespørgsel og dokumenter. Søgningen vil ofte forløbe som en interaktiv proces, hvor brugeren reformulerer forespørgslen ud fra en vurdering af de fremfundne dokumenter.

Vi vil referere til figur 2.1 flere gange i løbet af kapitlet for at relatere emnerne for de forskellige afsnit til den overordnede problemstilling - konstruktion og anvendelse af fuldtekst-søgesystemer. Kapitlet starter med et rids af den historiske udvikling indenfor forskningen i informationssøgning, derefter vender vi os mod de tekniske aspekter omkring lagring af og søgning i tekst. Kapitlet består af syv afsnit:

1. Historie
2. Lagringsteknikker
3. Tekstrepræsentation
4. Søgeteknikker
5. Faciliteter til at forbedre søgningen
6. Brugergrænsefladen
7. Sammenfatning

2.1 Historie

Udviklingen indenfor fuldtekstsøgning er en del af udviklingen indenfor det bredere område informationssøgning. I dette afsnit vil vi beskrive fuldtekstsøgning i den sammenhæng, det indgår i. Vi koncentrerer os om de overordnede aspekter og skift i udviklingen; afsnittet dækker således fuldtekstsøgning såvel som informationssøgning generelt.

I juli 1945 skrev Bush en artikel i *Atlantic Monthly*, hvor han lancerede ideen om Memex (Bush 1945):

> *Consider a future device for individual use, which is a sort of mechanized private file and library. It needs a name and, to coin one at random, 'memex' will do. A memex is a device in which an individual stores all his books, records and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility. It is an enlarged intimate supplement to his memory.*

Da artiklen udkom, var det ikke muligt at konstruere Memex; de nødvendige teknologier

eksisterede ikke. Det, Bush beskrev, var en vision. Artiklen skabte anseelig debat allerede ved udgivelsen, og Memex har lige siden tilhørt den gruppe af ikke-efterprøvede teorier, ubesvarede spørgsmål og ikke-byggede apparater, der har sat dagsordenen for talrige forskere (Smith 1981). Især blandt amerikanske forskere er det en udbredt opfattelse, at Bush's artikel markerer starten på forskningen indenfor edb-baseret informationssøgning og en række tilgrænsende forskningsområder.

Tolv år senere blev det næste store skridt i udviklingen af informationssøgesystemer taget, da Luhn (1957) kom med et konkret forslag til, hvordan man automatisk kunne afgøre, hvilke dokumenter der var relevante for en given forespørgsel. Han foreslog som den første, at dette skulle afgøres ved en sammenligning af statistisk bestemte indholdsidentifikatorer. Identifikatorerne - nøgleordene - skulle bestemmes ved optælling af frekvenserne for de ord, der forekom i dokumenterne.

Med Luhn's arbejde fik brugen af edb en central og etableret plads i informationssøgningen. Indtil da havde hjælpemidlerne i informationssøgningen alle været på bogform, fx forfatter- og emnekataloger. Siden da - begyndelsen af 60'erne - har udviklingen af informationssøgesystemer været baseret på to forskellige paradigmer, og et tredje er nu ved at etablere sig. Når vi i det følgende beskriver udviklingen ved hjælp af paradigmer, fokuserer vi på hovedstrømningerne. Bush (1945) og Mackay (1960) er eksempler på markante undtagelser; de har behandlet aspekter, som først længe efter er blevet taget op indenfor hovedstrømningerne. Beskrivelsen af de tre paradigmer - det system-drevne, det bruger-orienterede og det kognitive - bygger på (Ingwersen 1987) og (Ingwersen & Wormell 1990). En lignende opdeling findes i (Belkin & Vickery 1985).

Det system-drevne paradigme

Dette paradigme er det traditionelle paradigme indenfor informationssøgning. Det var dominerende før edb blev et redskab i informationssøgningen, og fra begyndelsen af 60'erne til ind i 70'erne var det også grundlaget for udviklingen af edb-baserede søgesystemer.

Indenfor dette paradigme fokuseres på tekstrepræsentation og søgeteknikker, dvs de faciliteter som udbyderen stiller til rådighed (se figur 2.1). Det er således effektiviteten og kvaliteten af selve søgesystemet, der er i centrum - heraf navnet 'system-dreven'. Brugeren ofres mindre interesse; det antages som oftest, at brugerens forespørgsel er en fuldstændig beskrivelse af den utilstrækkelige viden, der har givet anledning til forespørgslen.

En meget stor del af grundlaget for den senere forskning på området skabes i denne periode. Som eksempler kan nævnes term-vægtning, klynge-teknikkerne og relevans-feedback (dem vender vi tilbage til i afsnit 2.4 og 2.5). Udviklingen og evalueringen af disse teknikker skete efter naturvidenskabelige metoder, dvs ved kontrollerede laboratorie-eksperimenter og med udelukkende kvantitative metoder.

Fra starten er perioden præget af optimisme og gode resultater. Ingwersen udtrykker det med en anelse ironi (Ingwersen 1987):

> *It is an optimistic age. With little or no underlying theory the commercial online industry builds (and still does) large scientific IR systems in an ad-hoc way.*

Senere opstår der imidlertid problemer: Man kommer ikke afgørende videre. Der er to sider af dette problem: Dels er repræsentations- og søgeteknikkerne så veludviklede, at yderligere forskning kun kan forventes at give promilleforbedringer, dels vinder forskningsresultaterne ikke kommerciel udbredelse. I konsekvens af dette begynder nogle forskere at interessere sig for de ikke-tekniske aspekter af informationssøgningen.

Det bruger-orienterede paradigme

I løbet af 70'erne opstår det bruger-orienterede paradigme. Det sker som en reaktion på det tekniske fokus indenfor det system-drevne paradigme. Ingwersen (1987) bemærker, at en konsekvens af det system-drevne paradigme er, at brugerne forventes at tilpasse sig systemets krav. Det bruger-orienterede paradigme sætter fokus på de psykologiske og sociologiske aspekter. Ingwersen & Wormell (1990) mener, at det groft sagt kan udtrykkes sådan, at mens man indenfor det system-drevne paradigme retter hele opmærksomheden mod selve søgesystemet, så retter man indenfor det bruger-orienterede paradigme hele opmærksomheden mod brugeren.

Med det bruger-orienterede paradigme rettes opmærksomheden mod brugerens problem: Formaliseringen og repræsentationen af den utilstrækkelige viden (den venstre del af figur 2.1). Brugerens søgestrategier og lignende kvalitative variable bliver emner for videnskabelige undersøgelser. Samtidig ændres undersøgelserne fra laboratorie-eksperimenter til empiriske undersøgelser. Det skyldes, at laboratoriernes 'som om'-situationer er for kunstige til at give troværdige resultater, når det er søgeadfærd og lignende, der undersøges.

Hen igennem 70'erne var der meget få personsammenfald mellem forskere indenfor det bruger-orienterede paradigme og forskere indenfor det system-drevne paradigme. Forskerne bekendte sig i vid udstrækning til enten det ene eller det andet paradigme. Fra midten af 80'erne sker der imidlertid en tilnærmelse mellem de to paradigmer. Denne tilnærmelse skyldes for en stor del en fælles interesse for nogle af de teknikker, der er udviklet indenfor forskningen i kunstig intelligens.

Det kognitive paradigme

Fra midten af 80'erne er der flere og flere forskere, der overskrider grænsen mellem det system-drevne og det bruger-orienterede paradigme. Det har to årsager (Ingwersen & Wormell 1990): For det første hjælper hverken teknisk optimering eller forfining af bruger-modeller, hvis de ikke afprøves i praksis; og i sådanne afprøvninger må der nødvendigvis indgå både tekniske og psykologiske/sociologiske overvejelser. For det andet begynder mange forskere fra begge paradigmer - som nævnt ovenfor - at interessere sig for nogle af de teknikker, der er udviklet indenfor AI-forskningen.

Med det kognitive paradigme fokuseres ikke længere på en udvalgt del af informationssøgningen, men på hele situationen med en understregning af, at der er tale om en kognitiv proces. Det kognitive paradigmes indtog markeres fx af, at ordet 'intelligent' nu optræder ofte i litteraturen om informationssøgning (mest udtalt i betegnelsen 'intelligent information retrieval'). Med det nye paradigme er der opstået tre nye interesseområder for forskere i informationssøgning (Croft 1987): Ekspertsystemer, vidensrepræsentation og automatisk behandling af naturligt sprog. En væsentlig årsag, til at AI-teknikkerne inddrages, er ønsket om at udvikle informationssøgesystemer, hvor forespørgslerne kan stilles i naturligt sprog.

Endnu er det kognitive paradigme relativt nyt og har ikke givet ret mange konkrete resultater. Der er endnu først og fremmest tale om en ændring af holdninger og fokusering (Brooks 1987): Det nye paradigme har stimuleret interessen for, hvad intelligent søgeadfærd er, og hvad der skal til i tekstrepræsentation og søgeteknikker for, at systemerne kan simulere eller støtte intelligent søgeadfærd. Smith (1983) pointerer, at det i forbindelse med interessen for AI-forskningen er afgørende at skelne mellem to typer automatisering: Maskinin-telligens og maskinstøttet intelligens. Maskinintelligens svarer ifølge Smith til fuld automatisering; maskinen overtager opgaver, der hidtil blev udført af mennesker. Herved mister/afgiver mennesket kontrollen over og en del af indsigten i løsningen af opgaven. Ved maskin-støttet intelligens har mennesket den fulde kontrol over

løsningen af opgaven, men betjener sig af en (semiautomatisk) maskine. Her fastlægges grænserne for maskinens funktioner af, at intet må foregå bagom ryggen på brugeren.

Vores intention er at udvikle et søgesystem, der kan fungere som et redskab for fagfolk. Dette perspektiv svarer til maskinstøttet intelligens. Vi mener, man kan lave gode støttesystemer, men ikke intelligente systemer. Det forhindrer imidlertid ikke, at nogle af de teknikker, der er udviklet indenfor AI-forskningen, med fordel kan anvendes.

2.2 Lagringsteknikker

I dette afsnit beskrives sekventielle filer, inverterede filer og relationsdatabaser - de tre lagringsteknikker, der henholdsvis har været anvendt, bliver anvendt og med mellemrum påtænkes anvendt i forbindelse med fuldtekstsøgesystemer. Næsten alle eksisterende søgesystemer benytter inverterede filer og har et væsentligt fællestræk (Faloutsos & Chan 1988):

> *Text databases are traditionally large and have archival nature: There are insertions in them, but almost never deletions and updates.*

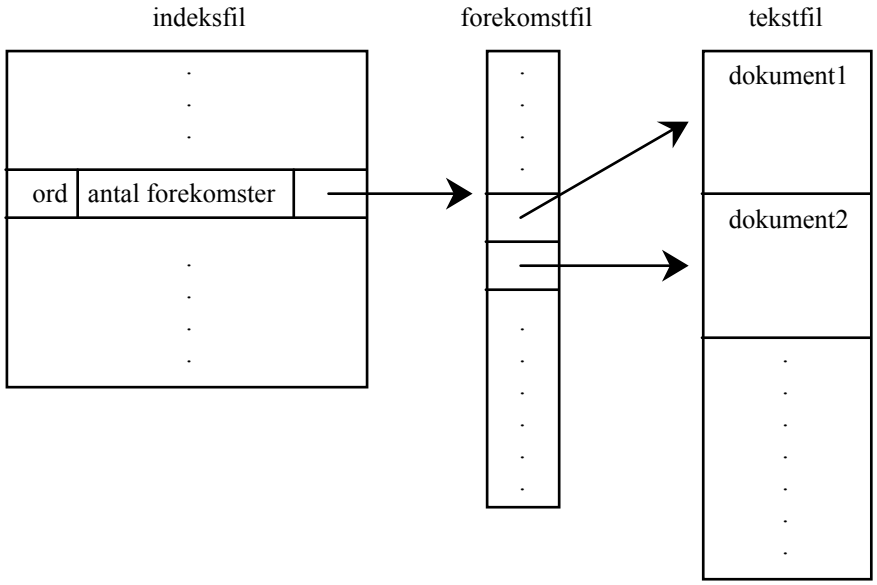
Dette fællestræk er næppe en begrænsning i forbindelse med fx bibliografiske databaser. I biblioteksverdenen antages det endvidere ofte, at indsættelser kan samles i bunker, så den medfølgende reorganisering kan foregå på få isolerede tidspunkter (Faloutsos 1985). Herved stilles beskedne krav til lagringsteknikken. Kontorverdenen, som er et af de væsentlige, nyere anvendelsesområder for fuldtekstsøgesystemer, er mere dynamisk. Faloutsos påpeger, at her må lagringsteknikken være mere fleksibel, specielt hvad angår rettelser og indsættelser.

Sekventielle filer og fuldtekstskanning

Den enkleste måde at lagre tekst på med henblik på søgning er at lagre teksten i en sekventiel fil og foretage søgningerne ved at skanne denne fil. Fordelene ved denne lagringsteknik er, at der ikke skal bruges anden lagerplads end den, der går til lagringen af teksten, og at den er meget let at implementere. Den afgørende ulempe er de lange søgetider. Fuldtekstskanning var effektivt dengang et stort antal forespørgsler blev behandlet samtidig i batch-kørsler, men er utilstrækkeligt nu, hvor forespørgslerne skal behandles enkeltvis og interaktivt (van Rijsbergen 1979). I de tilfælde hvor fuldtekstskanning bruges, udføres den som oftest ved hjælp af specialdesignet hardware, eller den kombineres med en anden lagringsteknik, fx inverterede filer. I en del søgesystemer foregår fremfindingen af de formodede relevante dokumenter således ved hjælp af inverterede filer, hvorefter der er mulighed for fuldtekst-skanning i disse dokumenter.

Inverterede filer

Inverterede filer er den mest udbredte afløser for sekventielle filer og fuldtekstskanning. Inverterede filer kan bruges både til nøgleordsbaseret søgning og til fuldtekstsøgning, idet inverteringen kan begrænses til invertering af nøgleord eller omfatte hele teksten. Ved brug af inverterede filer lagres dokumenterne i en sekventiel fil; ligesom det er tilfældet ved fuldtekstskanning. Forskellen er, at der ved siden af denne fil etableres en indeksstruktur.



Figur 2.2. Typisk filstruktur ved anvendelse af inverterede filer, efter Faloutsos (1985). Indeksfilen indeholder de inverterede ord, forekomstfilen angiver ordenes placering i teksten, og teksten ligger i en fil for sig.

Invertering svarer til oprettelse af et indeks. En inverteret fil består således af en indgang for hvert ord i teksten og hægter til alle ordets forekomster i teksten. Typisk opdeles en inverteret fil i to filer: En indeksfil og en forekomstfil (se figur 2.2). Adresseringen ind i indeksfilen kan foretages ved hjælp af B-træer, hashing eller lignende. Endnu en mulighed er at lave indeksfilen i flere niveauer (Faloutsos 1985). Ex: Ved en indeksfil i to niveauer indeholder det første niveau bogstavpar og adresserer ind i det andet niveau, hvor ord, der begynder med det samme bogstavpar, er lagret sammen. Inverterede filer har en række attraktive egenskaber; men der er nogle omkostninger forbundet med at opnå disse fordele (Faloutsos 1985):

Fordelene ved inverterede filer er, at:
- De er relativt lette at implementere.
- De er effektive, dvs søgningerne får korte svartider.
- De kan let udvides til at støtte håndtering af synonymer. Synonymringe kan etableres, blot der tilføjes et ekstra felt i indeksfilen med en hægte til det næste synonym i ringen.

Ulemperne ved inverterede filer er, at:
- De kræver meget lagerplads. Inverteringen medfører et ekstra forbrug af lagerplads på 50-300% i forhold til det, der går til lagringen af teksten.
- Det er tidskrævende at opdatere og reorganisere indekserne. Det er et problem, hvis der ofte rettes i teksten, ikke hvis den er statisk.

Mange søgesystemer giver brugeren mulighed for at slå op og bladre i indeksfilen (Tenopir & Ro 1990). Denne meget enkle facilitet kan yde brugeren en vis støtte i valget af søgeord, specielt hvis indeksfilen også indeholder synonymringe.

Relationsdatabaser

Relationsdatabaser blev introduceret af Codd i 1970 (Codd 1970). Relationsdatabasernes solide teoretiske fundament og elegante matematiske formulering har været anerkendt lige fra starten, se fx (Elliott 1971). Det varede imidlertid omkring 15 år, før relationsdatabaserne blev implementeret effektivt og begyndte at slå igennem kommercielt. Ved 70'ernes afslutning fandtes der således ingen kommercielt tilgængelige relationsdatabasesystemer. Flere systemer indeholdt visse relationelle ideer og aspekter; men de systemer, der søgte at realisere hele ideen, fx Ingres og System R, var stadig kun prototyper (Date 1986). I løbet af 80'erne etablerede relationsdatabaser sig som standarden for lagring og organisering af strukturerede data. Det er derfor nærliggende at overveje, om de også kan anvendes i forbindelse med tekst og andre ustrukturerede data.

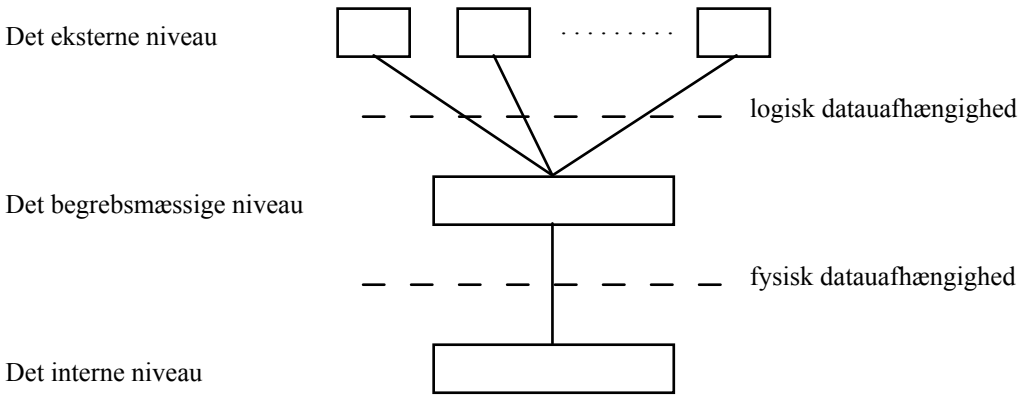
I flertallet af de eksisterende fuldtekstsøgesystemer er databasen baseret på inverterede filer. Forskellen på disse databaser og relationsdatabaser er en forskel i det abstraktionsniveau, brugeren/programmøren har mulighed for at betragte systemet på (Date 1986):

> *An inverted list database is similar to a relational database - but a relational data-base at a low level of abstraction, a level at which the stored tables themselves* ***and also certain access paths to those stored tables*** *(in particular, certain indexes) are directly visible to the user.* [fremhævning i originalen]

Flere relationsdatabasesystemer, fx DB2, er faktisk implementeret ved hjælp af inverterede filer. Abstraktionsniveauet er hele pointen ved relationsdatabaser: Brugerne/programmørerne tilbydes en enkel model - 2-dimensionelle tabeller hvor hver række er en n-tupel - samt et programmeringssprog til at manipulere med disse tabeller, og det er alt, hvad de behøver at bekymre sig om. Forbindelsen mellem tabellerne og de faktisk lagrede data er det relations-databasesystemets opgave at tage sig af. Det er således systemets opgave at lagre tabellerne på en hensigtsmæssig måde og at oversætte manipulationerne af tabellerne til ændringer i de lagrede data. I modsætning til andre systemer står og falder adgangen til de lagrede data således ikke med, om der eksisterer prædefinerede indeks eller lignende (Codd 1982). Vi kan sammenfatte relationsdatabasernes væsentligste fordele i to punkter:

- En enkel model kombineret med en konsistent og ensartet adgangsstruktur.
- Såvel logisk som fysisk datauafhængighed. Det er datauafhængigheden, der gør det muligt at betragte relationsdatabaser på forskellige abstraktionsniveauer.

En relationsdatabases artitektur kan opdeles i tre generelle niveauer (Date 1986): Det eksterne niveau, der omfatter den måde databasen fremstår på for den enkelte bruger; det begrebsmæssige niveau, der omfatter det, der er fælles for alle brugerne (den logiske database); og det interne niveau, der omfatter implementering og lagring (den fysiske database). Fysisk datauafhængighed er datauafhængighed mellem det interne og det begrebsmæssige niveau; logisk datauafhængighed er datauafhængighed mellem det begrebsmæssige og det eksterne niveau:



Figur 2.3. De tre niveauer i relationsdatabasers arkitektur og de to datauafhængigheder (niveaudelingen er generel, ikke specifik for relationsdatabaser). På grund af datauafhængigheden kan niveauerne betragtes uafhængigt af hinanden. Efter Date (1986).

Den fysiske datauafhængighed sikrer, at applikationerne ikke er afhængige af den måde, databasen er implementeret på. På det interne niveau er det eksempelvis muligt at erstatte inverterede filer med hash-adresserede filer uden, at det påvirker applikationerne. Den fysiske datauafhængighed omfatter tre punkter (Codd 1970):

- Uafhængighed af ordning. Der er ingen sammenhæng mellem den ordning, brugeren definerer af sine data, og den rækkefølge, dataene lagres i.
- Uafhængighed af indeks. Der skelnes skarpt mellem relationernes nøgler, der medvirker til at fastlægge dataenes mening, og indeks, der udelukkende bruges til at effektivisere databasen.
- Uafhængighed af adgangsstrukturen. Forespørgslerne skal formuleres deklarativt, ikke proceduralt.

Den logiske datauafhængighed betyder, at databasen i vid udstrækning kan udvides og omstruktureres uden, at de programmer, de enkelte brugere anvender, skal ændres. Dette muliggøres for det første af, at systemet har fuld kontrol over, hvordan og ad hvilke veje de lagrede data findes frem. Når databasen udvides eller omstruktureres, kan de nødvendige ændringer i den måde, de lagrede data findes frem på, derfor ske automatisk. Den anden grund til, at ændringer i databasen som regel kan gennemføres uden ændringer i brugernes programmer, er muligheden for at definere *views* - virtuelle tabeller. *Views* giver mulighed for at definere den enkelte brugers syn på databasen, dvs for at trække de data og sammenhænge frem, som er interessante for denne bruger. Hvis brugeren får behov for at se nogle andre data eller se data i en anden sammenhæng, kan der blot defineres et nyt *view*.

Udover de ovennævnte fordele ved relationsdatabaser opstiller Blair (1988) en liste med 13 yderligere fordele ved relationsdatabaser i forhold til eksisterende søgesystemer, rækkende fra detaljerede autorisationsregler til rapportgeneratorer. Til trods for disse fordele er der noget nær konsensus om, at inverterede filer er at foretrække for relationsdatabaser. Da de første relationsdatabaser kom frem i begyndelsen af 80'erne, var kritikken blandt andet rettet mod de to følgende punkter:

- Datastrukturerne er rettet mod strukturerede data, de støtter ikke lagring af tekst (Biller 1983). Codd (1982) er stort set enig. Fem år senere er udviklingen af relationsdatabaserne nået så meget længere, at Ashford (1987) mener, datastrukturerne til behandling af data med stor og varierende længde er forbedret afgørende.
- Mange hyppigt forekommende forespørgsler er besværlige at formulere i SQL, fx (Crawford 1981). Dette problem forsvinder efterhånden, som mere avancerede bruger-

grænseflader udvikles. SQL skal jo kun bruges internt i systemet; det er ikke meningen, brugerne skal formulere deres forespørgsler direkte i SQL.

Nu er kritikken af relationsdatabaserne rettet mod deres ressourceforbrug: Ulemperne ved relationsdatabaser er, at de kræver for meget lagerplads udover det, der går til lagringen af selve teksten, og at typiske forespørgsler udføres for langsomt. Lynch & Stonebraker (1988) præciserer disse to punkter:

- Der bruges meget plads på at lagre redundant information. Redundansen forekommer i forbindelse med relationernes nøgler: For det første forekommer den samme nøgle typisk som nøgle i flere relationer og lagres derved gentagne gange. For det andet lagres nøglerne - i nogle tilfælde - både i tuplerne og i det indeks, der hører til relationen.
- Der bruges meget tid på *joins*. For at opfylde de forskellige normalformers krav består relationsdatabaser af mange relationer. En typisk forespørgsel omfatter derfor flere *joins*, og hver *join* er en tidskrævende operation.

Det store pladsforbrug er - som allerede nævnt - også et problem i forbindelse med inverterede filer; men problemet er endnu større med relationsdatabaser, hvor nøgler, der forbinder to relationer, lagres i begge relationerne. De lange svartider ses primært i forbindelse med relationsdatabaser og skyldes problemer med at modellere dokumentstrukturen effektivt ved hjælp af relationer. Der er gjort flere forsøg på at undgå de mange *joins* blandt andet ved at bruge databaser, der ikke er på første normalform, fx (Jaeschke & Schek 1982). Lynch & Stonebraker (1988) har en anden, men lignende, idé; de erstatter *joins* med operatorer. Vi vil beskrive Lynch & Stonebraker's forslag ud fra et eksempel:

Et fuldtekstsøgesystem kan med fordel indeholde felter med oplysninger om dokumenternes forfattere og lignende. Men da et dokument kan have flere forfattere, må der oprettes både en dokument-relation og en forfatter-relation, hvis databasen skal være på første normalform. Når en bruger beder om et dokument skrevet af en bestemt forfatter, må der således foretages en *join* af forfatter-relationen og dokument-relationen. Lynch & Stonebraker's alternativ er at samle alle forfatternavnene i en streng og lade den være et felt i dokument-relationen. Derudover implementeres en operator, der kan trække de enkelte forfatternavne ud af forfatter-feltet. Lynch & Stonebraker's eksperimenter indikerer, at denne fremgangsmåde er betydeligt mere effektiv end brugen af *joins*.

2.3 Tekstrepræsentation

I fuldtekstsøgesystemer benyttes fuldtekstrepræsentation. Alternativet til dette er indeksering, hvor dokumenterne repræsenteres ved nøgleord. I dette afsnit redegøres for nogle resultater, som er nået i forbindelse med indeksering, men også er relevante i forbindelse med fuldtekstrepræsentation, og fuldtekstrepræsentation og indeksering sammenlignes. Vi begynder imidlertid med en - meget kort - beskrivelse af, hvad indeksering er.

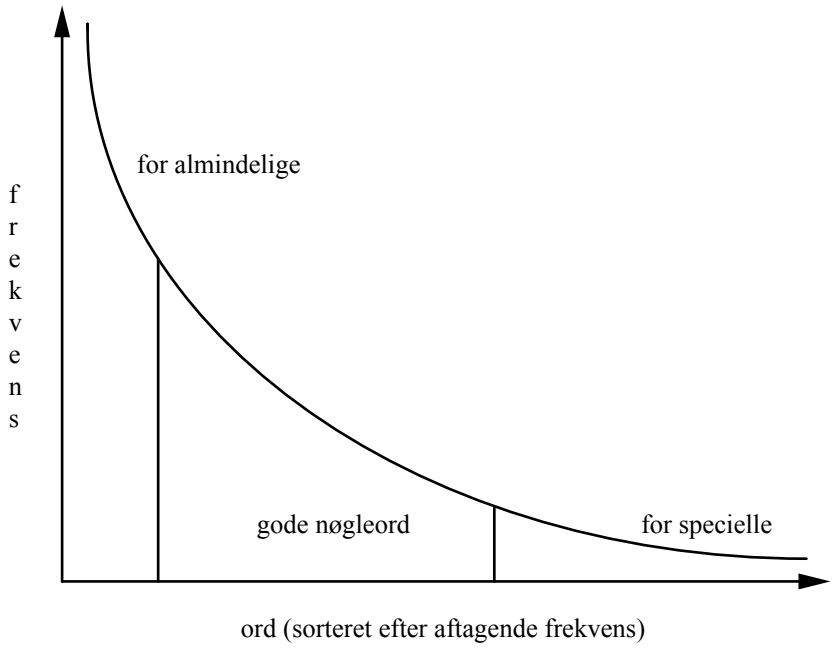
Formålet med indekseringen er at muliggøre effektiv søgning. Indekseringen skal således frembringe præcise og specifikke nøgleord, der trækker det unikke i hvert dokument frem. Et dokuments nøgleord skal karakterisere dokumentet og adskille det fra de andre dokumenter. Indeksering kan foregå manuelt/intellektuelt ved, at en indeksør gennemlæser dokumentet og afgør, hvilke nøgleord der er relevante. I så fald vælges nøgleordene næsten altid fra en tesaurus eller et andet kontrolleret glosarium. Nøgleordene behøver således ikke forekomme i dokumenterne. Indekseringen kan også foregå automatisk ud fra optællinger af ordfrekvenser, se fx (Sparck Jones 1974), (Salton

& McGill 1983) og (Salton & Buckley 1988). I dette tilfælde vælges nøgleordene ikke fra et kontrolleret glosarium; de kommer fra dokumenterne.

Velegnede nøgleord

Hvis nøgleordene ikke skal tages fra et prædefineret glosarium, skal det være muligt at genkende gode nøgleord, når 'man ser dem'. Dette problem er blevet indgående behandlet i forbindelse med automatisk indeksering.

Automatisk bestemmelse af nøgleord er baseret på optællinger af, hvor hyppigt ordene forekommer i de enkelte dokumenter. Et ords frekvens angiver således, hvor mange gange det forekommer i et givet dokument. Luhn (1958) introducerede ideen om en øvre og en nedre grænse for gode nøgleords frekvens. Kvantificering af disse grænser er naturligvis meget usikker; i (Salton m.fl. 1975) foreslås en frekvens på 1% af dokumentets ord som nedre grænse og 10% som øvre grænse. De to grænser opdeler ordene i dokumentet i tre grupper:



Figur 2.4. Sammenhængen mellem et ords frekvens og dets værdi som nøgleord, efter (van Rijsbergen 1979).

Ordene med høj frekvens er dårlige nøgleord; de er for almindelige og siger derfor ikke noget om dokumentets indhold. Ordene med lav frekvens er ligeledes dårlige nøgleord; de er for specielle og vil derfor sjældent blive brugt i forespørgsler. Ordene med frekvenser i mellemområdet er derimod meningsbærende og gode nøgleord.

Denne tredeling af ordene i et dokument kan ikke blot bruges til automatisk indeksering; den er også værdifuld i forbindelse med fuldtekstsøgning. Ordene med lav frekvens kan ikke bruges som søgeord direkte; men de har måske et synonym eller et overbegreb, der er mere almindeligt. De kan derfor ofte transformeres til gode søgeord ved hjælp af en tesaurus (se afsnit 2.5) (Crouch 1988).

Ordene med høj frekvens er som regel ikke noget bevendt som søgeord og vil fylde meget i den inverterede fil; de skal derfor enten placeres i stoplisten eller inkluderes i vendinger. En stopliste er en negativliste; en liste over ord, der ikke har nogen værdi som søgeord, og derfor skal filtreres fra under inverteringen. Ifølge Fox (1989) kan stoplister ikke blot genereres automatisk ud fra ordenes frekvenser. En manuel kontrol er

nødvendig, da enkelte ord med høj frekvens er gode søgeord. Typiske elementer i en stopliste er ord som 'og', 'en' og 'at'. En vending består af flere ord, men udgør én samlet meningsenhed, fx 'eksakt match teknikker' og 'virtuelle tabeller'. Ord, der er dårlige søgeord på grund af deres høje frekvens, kan muligvis inkluderes i vendinger, som på grund af deres snævrere betydning og lavere frekvens er gode søgeord.

Mål for indekseringsglosarier og søgeeffektivitet

De to traditionelt mest anvendte mål for indekseringsglosariers effektivitet er *exhaustivity* og *specificity* (van Rijsbergen 1979). Vi oversætter disse to ord til glosariets bredde henholdsvis dets dybde. Glosariets bredde siger noget om det antal forskellige emner, der kan indekseres med glosariet. Glosariets dybde siger noget om glosariets evne til at beskrive emnerne præcist - hvor detaljeret glosariet er. Begge disse mål er meget svære at kvantificere. Det er blevet foreslået (Sparck Jones 1973), at et glosariums bredde er relateret til antallet af nøgleord, der hæftes på et givet dokument, og at dets dybde er relateret til antallet af dokumenter, der får et givet nøgleord hæftet på sig - få dokumenter svarer til stor dybde. På grund af problemerne med at kvantificere disse mål vurderes indekseringsglosarier imidlertid ofte ud fra effektiviteten af søgninger i dokumentsamlinger, der er indekseret med glosariet.

En søgning bør ideelt set fremfinde samtlige relevante dokumenter og kun dem. En søgnings effektivitet afgøres således blandt andet af, hvor stor en del af de fremfundne dokumenter, der er relevante. Det giver et mål for søgesystemets evne til at afvise irrelevante dokumenter. Et andet forhold med afgørende betydning for søgeeffektiviteten er, i hvor høj grad alle relevante dokumenter findes frem. Vurderingen af et søgesystems søgeeffektivitet afhænger således af to parametre, dels mængden af fremfundne henholdsvis ikke-fremfundne dokumenter, dels mængden af relevante henholdsvis ikke-relevante dokumenter:

	relevante	ikke-relevante	ialt
fremfundne	$A \cap B$	$\overline{A} \cap B$	B
ikke-fremfundne	$A \cap \overline{B}$	$\overline{A} \cap \overline{B}$	$\overline{B}$
ialt	A	$\overline{A}$	N

Figur 2.5. Sammenhængen mellem relevans/ikke-relevans og fremfundne/ikke-fremfundne dokumenter (van Rijsbergen 1979). N er det samlede antal dokumenter i systemet, A er det samlede antal relevante dokumenter, og B er det samlede antal fremfundne dokumenter. Det skal bemærkes, at kvantificeringen af A er behæftet med stor usikkerhed.

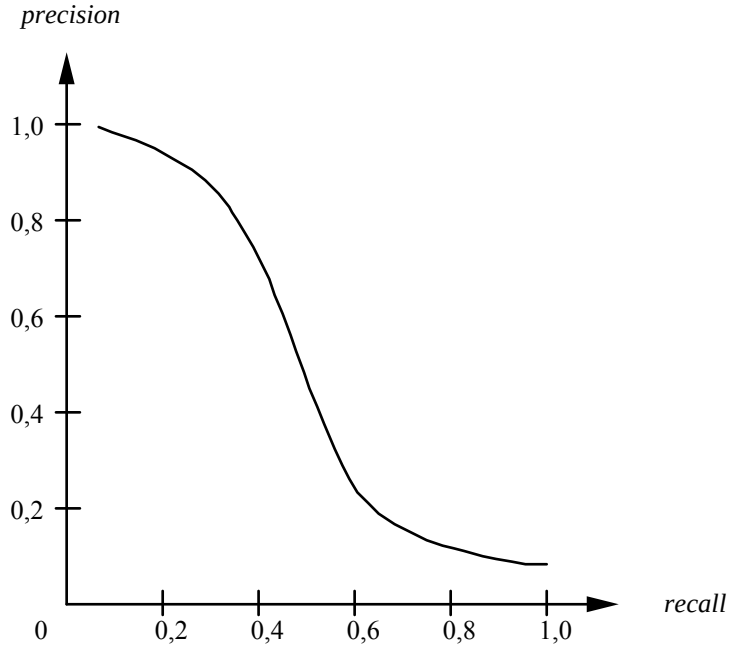
Figur 2.5 giver mulighed for at definere en række forskellige mål for søgeeffektivitet. Vi vil kun komme ind på de to mest udbredte, *recall* og *precision*. *Recall* måler i hvor høj grad, de relevante dokumenter findes frem, mens *precision* måler i hvor høj grad, de irrelevante dokumenter afvises. På baggrund af figuren er *recall* og *precision* defineret ved:

$$recall = \frac{|A \cap B|}{|A|}$$

$$precision = \frac{|A \cap B|}{|B|}$$

Recall er således den brøkdel af de relevante dokumenter, der findes frem, og *precision* er den brøkdel af de fremfundne dokumenter, der er relevante. Selvom *recall* og *precision* er lettere at kvantificere og forholde sig til end indekseringsglosariets bredde og dybde, er de ikke uden problemer. Det er problematisk at kvantificere det totale antal relevante dokumenter, A, og derfor er kvantificeringen af *recall* usikker. *Recall* og *precision* er desuden baseret på en grov forenkling: Et dokument er enten relevant eller irrelevant. I praksis er mange dokumenter delvis relevante. Blair & Maron (1985) har således i en empirisk undersøgelse mødt en opdeling af de fremfundne dokumenter i fire grupper: De irrelevante, de marginalt relevante, de tilfredsstillende og de vitale. Denne opdeling blev introduceret af de jurister, der deltog i undersøgelsen. Det var en opdeling, de var vant til at arbejde med. Der er naturligvis stor forskel på at gå glip af et vitalt og et marginalt relevant dokument.

Det ideelle er naturligvis at opnå både et *recall* og en *precision* på 100%. Det betragtes imidlertid som fastslået, at der i praksis altid bliver tale om en afvejning (se figur 2.6). De brugere, der er afhængige af et højt *recall*, må således være indstillede på at skulle formulere deres forespørgsler så bredt, at de fremfundne dokumenter også omfatter en masse irrelevante. Omvendt indeholder en søgning, der næsten udelukkende resulterer i relevante dokumenter, også en indikation af, at der er mange relevante dokumenter tilbage blandt de ikke fremfundne.



Figur 2.6. Typisk graf over forholdet mellem *recall* og *precision*, efter (Salton & McGill 1983). Grafen viser naturligvis et gennemsnit over mange søgninger.

Sammenligning af indeksering og fuldtekstrepræsentation

I det følgende sammenlignes indeksering - det være sig manuel eller automatisk - og

fuldtekstrepræsentation. Tenopir & Ro (1990) refererer resultaterne fra en lang række forskningsprojekter. Der er nogenlunde enighed om, at fuldtekstrepræsentation giver højere *recall* end indeksering. Med hensyn til *precision* er enigheden mindre udtalt. En del undersøgelser viser, at indeksering resulterer i højere *precision* end fuldtekstrepræsentation; andre finder, at de to metoder er ret ens, hvad *precision* angår. Tenopir & Ro konkluderer, at valget mellem fuldtekstsøgning og nøgleordsbaseret søgning afhænger af, hvad søgningen skal give. Hvis brugeren ønsker få, meget relevante dokumenter, er nøgleordsbaseret søgning som regel at foretrække. Hvis brugeren ønsker mange og også partielt relevante dokumenter, er fuldtekstsøgning som regel at foretrække.

Lancaster m.fl. har i en undersøgelse af *Epilepsy Abstracts Retrieval System* sammenlignet søgning på nøgleord og søgning på såvel nøgleord som resuméer (et skridt i retning af fuldtekstsøgning). De finder markant højere recall, når resuméerne inddrages i søgningen, og angiver blandt følgende årsager (Lancaster m.fl. 1972):

> *- The statistical probability of getting a "hit" with the abstract and index terms is greater because there are more words.*
> *- The same concept can be represented in various ways in the abstract, so if one strategy does not match the terms used, another may.*
> *- To get a hit with the index terms alone requires the use of the precise terms used in the thesaurus.*
> *- The natural language of the abstract seems to match the language of the searchers better than the language of the index terms.*
> *- Indexing errors occur in Epilepsy Abstracts.*
> *- Epilepsy Abstracts is not fully consistent in its indexing, and there is considerable overlap among some of the terms.*

Disse årsager er efter vores mening en god sammenfatning af begrundelserne for at bruge fuldtekstsøgning. I 1985 stillede Blair & Maron imidlertid et alvorligt spørgsmålstegn ved fuldtekstsøgesystemernes effektivitet (Blair & Maron 1985), (Blair 1986).

Blair & Maron udførte en omfattende serie eksperimenter, hvor jurister brugte et fuldtekstsøgesystem til at finde det materiale, de fandt relevant for en række hypotetiske sager. Juristerne var afhængige af et meget højt *recall*. De kunne ikke tillade sig at overse et vitalt dokument; men de var også villige til at acceptere en tilsvarende lavere *precision*. Juristerne kvantificerede deres krav til et *recall* på 75% og blev bedt om at søge indtil, de var tilfredse med udbyttet. Det forbløffende resultat var et *recall* på 20% og en *precision* på 79%, i gennemsnit.

Juristerne troede altså, de havde fundet 75% af de relevante dokumenter, men havde reelt kun fundet 20%. En forklaring på dette kunne være, at juristerne forvekslede *recall* og *precision*. Blair & Maron afviser imidlertid denne hypotese - den er ikke statistisk signifikant. De konkluderer i stedet, at det er yderst svært - hvis ikke umuligt - for brugerne at forudse hvilke ord og vendinger, der kendetegner blot flertallet af de relevante dokumenter. De sammenfatter årsagerne til, at ordvalget er så uforudsigeligt, i en række punkter blandt andet de følgende fem (Blair & Maron 1985):

- I nogle relevante dokumenter bruges centrale begreber og vendinger overhovedet ikke. Ex: Et dokument kan udelukkende benytte navne på specifikke produkter uden at nævne deres bredere fællesbetegnelse. Da der kan være hundredevis af sådanne produkter, kan en tesaurus ikke altid klare problemet.
- Den synsvinkel, der lægges på sagen, er ofte afgørende for ordvalget. Forskellige (grupper af) dokumenter vil således ofte omtale den samme sag med forskellige ord. Ex: Mens politiet taler om 'et færdselsuheld', taler de pårørende om 'en tragedie', og

forsvareren om 'en situation'.

- Navne på teknikker, begreber og lignende deles ofte kun af en lille gruppe personer. Søgning på tværs af disse grupper kræver derfor dybtgående kendskab til det pågældende emne. Ex: Blair & Maron refererer, at juristerne i deres eksperimenter brugte søgeordet "trap correction". Andre generelle navne for dette viste sig at være "wire warp" og "the shunt correction system"; opfinderen af systemet omtalte det imidlertid - som den eneste - som "the Roman circle method", og i en anden by, hvor systemet også havde været testet, havde det fået det lokale navn "air truck".
- Brugen af slang er endnu en årsag til forskelle i ordvalg.
- Stave- og slåfejl kan måske ikke netop betegnes som forskelle i ordvalg; men de giver de samme problemer under søgning og er endnu mere uforudsigelige.

Ved søgning i store dokumentsamlinger kan en forespørgsel ikke blot bestå af et enkelt ord, idet en sådan søgning finder for mange dokumenter frem. I store dokumentsamlinger må forespørgslerne derfor bestå af konjunktioner af flere søgeord. Uforudsigeligheden af ordvalget i dokumenterne betyder imidlertid, at det er problematisk at indsnævre søgningen ved at tilføje andre ord. Et eksempel (Blair & Maron 1985):

> $P(Bt_i)$ betegner sandsynligheden for at brugeren bruger termen t_i.
> $P(Dt_i)$ betegner sandsynligheden for at t_i forekommer i et relevant dokument.
>
> To termer, t_1 og t_2, resulterer i, at et relevant dokument findes frem, hvis de både indgår i forespørgslen og et relevant dokument. Da ordvalget er svært at forudsige, er det ikke urimeligt at sætte $P(Bt_1) = 0,6$, $P(Bt_2) = 0,5$, $P(Dt_1) = 0,7$ og $P(Dt_2) = 0,6$.
>
> Sandsynligheden for, at brugeren bruger t_1 eller t_2 og et relevant dokument indeholder denne term, er således:
>
> t_1: $P(Bt_1) * P(Dt_1) = 0,42$
> t_2: $P(Bt_2) * P(Dt_2) = 0,30$
>
> Sandsynligheden for, at brugeren bruger såvel t_1 som t_2 og et relevant dokument indeholder begge termer, er derimod:
>
> t_1 og t_2: $P(Bt_1) * P(Dt_1) * P(Bt_2) * P(Dt_2) = 0,12$
>
> Man må altså forvente drastiske fald i *recall*, når forespørgslerne opbygges ved konjunktion af flere ord.

Undersøgelsen leder Blair til en konklusion, som er en direkte afvisning af, hvad forskerne indtil da har været nogenlunde enige om (Blair 1986):

> *[...] the study should put to rest any lingering belief in the potential for simple full-text retrieval as a method for gaining high Recall and tolerable Precision in searching large document data bases.*

Blair & Maron's undersøgelse fik imidlertid ikke lov at stå uimodsagt ret længe. Salton (1986) argumenterer således eksplicit for, at de ikke har ret i deres konklusioner. Saltons hovedargument er, at der er en snæver og uomgængelig sammenhæng mellem *recall* og *precision*, højt *recall* betyder lav *precision* og omvendt (se figur 2.6). Med dette som udgangspunkt finder han ikke Blair & Maron's resultater usædvanlige. Flere andre undersøgelser har vist, at en *precision* på omkring 80% giver et *recall* på ca 20%. Blair & Marons resultater er således i god overensstemmelse med den sammenhæng mellem *recall*

og *precision*, som er fundet i tidligere empiriske undersøgelser og illustreret i figur 2.6. Forespørgslerne i Blair & Maron's eksperimenter skulle altså have været væsentligt bredere, så havde de formodentlig givet de ønskede resultater. Saltons kritik af Blair & Maron's konklusioner betyder, at deres undersøgelse ikke kan bruges til at afvise fordelene ved fuldtekstsøgning. Salton anfægter imidlertid ikke de resultater, Blair & Maron baserer deres konklusioner på. Hans kritik skal derfor ikke bruges til at afvise de ulemper ved fuldtekstsøgning, som Blair & Maron beskriver.

2.4 Søgeteknikker

I dette afsnit gives en oversigt over søgeteknikker, der anvendes i forbindelse med fuldtekstsøgning. Afsnittet behandler således den midterste del af figur 2.1. Vi vil beskrive boolsk søgning med nærhedsoperatorer, udvidet boolsk søgning og skimming.

Boolsk søgning og nærhedsoperatorer

Boolsk søgning er standarden for fuldtekstsystemer. Kommercielt er boolsk søgning så at sige enerådende, og i forskningen er den også udbredt. Ved boolsk søgning består forespørgslerne af søgeord sammensat ved hjælp af de logiske operatorer. Fortolkningen af forespørgslerne er baseret på mængder og mængdeoperatorer (Bookstein 1981). Til hvert søgbart ord, dvs hvert ord i den inverterede fil, er knyttet en mængde: Mængden af ordets forekomster. De logiske operatorer svarer til hver deres mængdeoperator:

OG	svarer til	fællesmængden
ELLER	svarer til	foreningsmængden
IKKE	svarer til	komplementærmængden

Styrken ved boolsk søgning er, at den giver brugerne et udtryksfuldt sprog, hvor de præcist og detaljeret kan angive den logiske sammenhæng mellem forespørgslens elementer. Det er endvidere let at foretage ændringer i forespørgslen, hvis den viser sig at være skæv i forhold til informationsbehovet (Bookstein 1983). Der er imidlertid også en række svagheder ved boolsk søgning; de fem væsentligste er:

- Det er en eksakt match teknik. Det er kun de dokumenter, der opfylder forespørgslen fuldstændigt, som findes frem.
- Der tages ikke hensyn til ordenes relative vigtighed. Det er ikke muligt at tildele et ord i forespørgslen særlig vægt. Ordene i dokumenterne betragtes ligeledes som lige vigtige.
- Boolsk søgning resulterer ikke i nogen rangordning af de fremfundne dokumenter. Dokumenterne deles blot i to grupper: De fremfundne og de ikke-fremfundne.
- Forespørgslerne skal formuleres som komplicerede, formelle udtryk. Brugerne skal ikke blot formalisere og repræsentere den utilstrækkelige viden; de skal formulere den i et sprog, der for de flestes vedkommende ligger langt fra det naturlige sprog, de er vant til at udtrykke sig i.
- Det er kun muligt at søge på enkeltord. Søgning på vendinger må formuleres som konjunktioner, fx 'eksakt OG match' for 'eksakt match'. Konjunktionen sikrer imidlertid kun, at ordene forekommer i samme dokument, ikke at de udgør en vending.

Der er gjort mange forsøg på at komme ud over en eller flere af disse svagheder, enten ved at anvende helt andre søgeteknikker eller ved at udvide/kombinere boolsk søgning med andre teknikker. Tenopir & Ro (1990) foreslår en meget simpel mulighed: Ved præsentationen af de fremfundne dokumenter angives for hvert dokument, hvor mange gange søgeordene forekommer i dokumentet. I det følgende vil vi behandle to

mere vidtgående muligheder - nærhedsoperatorer og begrebsbaseret fremfinding.

Den mest almindelige udvidelse af boolsk søgning er nærhedsoperatorer. Nærhedsoperatorer giver brugeren mulighed for at angive, at to ord ikke blot skal forekomme i samme dokument, men i en snævrere defineret kontekst, fx samme afsnit eller samme sætning. Nærhedsoperatorerne ligner således OG-operatoren: Ved OG-operatoren er det underforstået, at konteksten altid er 'indenfor samme dokument'; med nærhedsoperatorer bliver det muligt at variere denne kontekst. En særlig nærhedsoperator er den, hvor konteksten er 'ved siden af'. Denne nærhedsoperator muliggør søgning på vendinger:

NABO	'eksakt NABO match' finder et dokument frem, hvis vendingen 'eksakt match' forekommer i dokumentet.

Nærhedsoperatorer bruges ud fra en antagelse om, at der er en tættere betydningsmæssig sammenhæng mellem ord, der står i nærheden af hinanden, end mellem ord, der står langt fra hinanden. Hensigten med nærhedsoperatorer er at give brugeren en vis mulighed for at angive den indbyrdes sammenhæng mellem søgeordene. I forhold til OG-operatoren bruges nærhedsoperatorer således til at afvise nogle af de dokumenter, hvor søgeordene nok forekommer, men uden den indbyrdes sammenhæng, brugeren er interesseret i. Nærhedsoperatorer bruges altså i håb om at øge søgningernes *precision*. Det understreges af, at de dokumenter, der findes frem ved brug af en nærhedsoperator, altid er en delmængde af de dokumenter, der findes frem med OG-operatoren eller en nærhedsoperator med en bredere kontekst. Hvis brugerne er afhængige af et højt *recall*, kan de således med fordel begrænse brugen af nærhedsoperatorer.

Nærhedsoperatorer kan implementeres ved at skanne teksten omkring det ene ord for at se, om det andet ord forekommer; det er imidlertid tidskrævende. Normalt implementeres nærhedsoperatorer ved at udvide den inverterede fil med oplysninger om ordenes position i teksten (Salton & McGill 1983). Det kan fx gøres ved at knytte et afsnitsnummer, et sætningsnummer og et ordnummer til hver ordforekomst. Ex: (1, 2, 3) betyder, at ordet forekommer som det 3. i den 2. sætning i dokumentets 1. afsnit.

En anden mulighed for at bringe boolsk søgning ud over nogle af sine begrænsningerne er begrebsbaseret fremfinding (Bing 1981). Begrebsbaseret fremfinding er blot en brugergrænseflade, der lægges ovenpå boolsk søgning. Begrebsbaseret fremfinding giver brugeren mulighed for at formulere sine forespørgsler uden at skulle bekymre sig om de boolske operatorer. Kernen i begrebsbaseret fremfinding er, at forespørgslerne formuleres som en fællesmængde af begreber. Ex: En funktionær vil have noget at vide om sine lønforhold under ferie. Det første skridt i formuleringen af forespørgslen er at strukturere den utilstrækkelige viden som en fællesmængde af tre begreber: Funktionærer, løn og ferie.

Ved begrebsbaseret fremfinding angives en forespørgsel ved, at brugeren beskriver hvert begreb med et eller flere ord. Brugeren starter indtastningen af forespørgslen, hvorefter systemet beder om de ord, der beskriver det første begreb. Når det første begreb er beskrevet, markerer brugeren, at forespørgslen er en fællesmængde af dette begreb og endnu et. Derefter beder systemet om de ord, der beskriver det næste begreb. Denne proces gentages for alle de begreber, forespørgslen er en fællesmængde af. Ex: Funktionæren fra før beskriver begrebet løn med ordene 'løn', 'gage' og 'salær'.

Ud fra brugerens forespørgsel, genererer systemet en forespørgsel formuleret med de boolske operatorer, og denne boolske forespørgsel udføres. Sammenhængen mellem den forespørgsel, brugeren angiver, og den, systemet genererer, er meget simpel, så brugen af begrebsbaseret fremfinding kan synes overflødig. Fordelen er imidlertid, at den måde,

forespørgslen angives på, støtter brugeren ved at bidrage til at strukturere den utilstrækkelige viden. Formuleringen af forespørgslen kan derfor ske på et abstraktionsniveau, der er tættere på brugerens, end det er tilfældet ved almindelig boolsk søgning.

Udvidet boolsk søgning

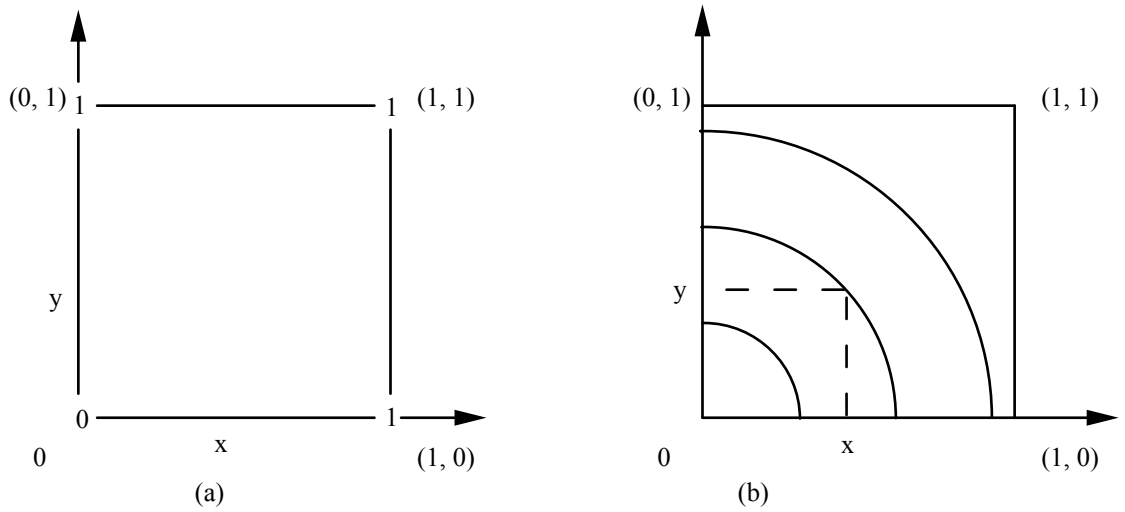
Der er gjort flere forsøg på at udvide boolsk søgning til en partiel match teknik. Et eksempel er fuzzy mængder, se fx (Bookstein 1981). Et andet og mere succesfuldt forsøg går under navnet udvidet boolsk søgning (Belkin & Croft 1987). Udvidet boolsk søgning, der her beskrives ud fra (Salton m.fl. 1983), diskuteres en del i litteraturen, men har ikke vundet kommerciel udbredelse. Udvidet boolsk søgning er en kombination af boolsk søgning og term-vægtning, der er en partiel match teknik. Formålet med kombinationen er at få både den struktur, der kendetegner forespørgslerne i boolsk søgning, og muligheden for at arbejde med vægte.

Ved udvidet boolsk søgning kan et ord indgå i en forespørgsel med en vilkårlig vægt mellem 0 og 1. En forespørgsel med n ord kan således betragtes som et punkt i et n-dimensionelt rum, hvor hver dimension svarer til et ord. Det er ikke blot søgeordene, der kan knyttes vægte til. Det er også muligt at knytte vægte til de boolske operatorer og ordene i dokumenterne. I forbindelse med fuldtekstsøgning kan mulighederne for at knytte vægte til søgeordene og de boolske operatorer bruges umiddelbart, mens muligheden for at knytte vægte til ordene i dokumenterne forudsætter, at oplysninger om ordenes frekvenser i de enkelte dokumenter er til rådighed. Vægtningen af ordene i dokumenterne indskrænkes derfor ofte til en angivelse af, om ordet forekommer i dokumentet eller ej. Et eksempel på en udvidet boolsk forespørgsel (tallene er de tilknyttede vægte):

tesaurus(1) OG(5) (fuldtekst(0,5) ELLER(2) fritekst(0,5))

Et søgeords vægt bruges til at angive, hvor vigtigt det er, at dette ord forekommer i de dokumenter, der findes frem. Vægtene på de boolske operatorer bruges til at angive, hvor vigtigt det er, at ordene i dokumenterne forekommer i kombinationer, der opfylder den boolske operator. Ved OG-operatoren angiver vægten således om, det er et ufravigeligt krav, at begge søgeordene forekommer, eller om det blot er at foretrække frem for dokumenter, der kun indeholder ét af de to søgeord.

Indenfor udvidet boolsk søgning opereres kun med to logiske operatorer, OG og ELLER. To søgeord med tilknyttede vægte kan betragtes som et punkt, (x, y). ELLER-operatoren defineres som ordningen: Des større afstand fra (x, y) til origo des bedre (se figur 2.7 (b)). Herved adskiller udvidet boolsk søgning sig fra boolsk søgning, hvor ELLER-operatoren alene defineres ud fra de fire mulige kombinationer af, hvorvidt de to søgeord forekommer eller ikke forekommer i dokumenterne (se figur 2.7 (a)). På samme måde defineres OG-operatoren som ordningen: Des mindre afstand fra (x, y) til (1, 1) des bedre. Dette adskiller sig fra boolsk søgning, hvor OG-operatoren kun er opfyldt, når (x, y) er lig med (1, 1).



Figur 2.7. Forespørgslen x ELLER y. (a) Ved boolsk søgning. Det er kun kvadratets hjørner, der benyttes: Et ord er enten med eller ikke med i forespørgslen og ELLER-operatoren er enten opfyldt eller ikke opfyldt. (b) Ved udvidet boolsk søgning. Ordene kan indgå i forespørgslen med en vilkårlig vægt mellem 0 og 1, og ELLER-operatoren defineres ved afstanden fra (x,y) til (0,0).

Det benyttede afstandsmål er L_p-vektornormen, der er en simpel generalisering af den sædvanlige euklidiske afstand. Afstandsmålet normaliseres for at gøre det mere bekvemt. For en vektor $\mathbf{v} = (v_1, ..., v_n)$, hvor alle koordinater er mellem 0 og 1, er den normaliserede L_p-vektornorm givet ved:

$$L_p(\mathbf{v})_{norm} = \frac{(v_1^p + ... + v_n^p)^{1/p}}{\quad}$$

Hvis ordenes vægte i dokumenterne betegnes d_i ($0 \leq d_i \leq 1$), og ordenes vægte i forespørgslen betegnes t_i ($0 \leq t_i \leq 1$), så defineres OG og ELLER ved følgende forespørgsel/dokument-ligheder ($1 \leq p \leq \infty$):

$$sim(D, Q_{OG(p)}) = 1 - \frac{(t_1^p(1 - d_1)^p + ... + t_n^p(1 - d_n)^p)^{1/p}}{(t_1^p + ... + t_n^p)^{1/p}}$$

$$sim(D, Q_{ELLER(p)}) = \frac{(t_1^p d_1^p + ... + t_n^p d_n^p)^{1/p}}{(t_1^p + ... + t_n^p)^{1/p}}$$

Værdien af p er vægten af den logiske operator. Man kan vise, at for $p = 1$ svarer udvidet boolsk søgning fuldstændig til term-vægtning, og for $p = \infty$ svarer den fuldstændig til boolsk søgning, se (Salton m.fl. 1983). For $1 < p < \infty$ fås en kombination. Brugeren kan således bruge vægte på de boolske operatorer til at stille krav. Et krav om, at to givne ord begge forekommer i de dokumenter, der findes frem, udtrykkes ved en OG-operator med vægten ∞. En ELLER-operator med vægt ∞ angiver et krav om, at mindst ét af de to involverede ord skal forekomme; det gør ingen forskel om det ene, det andet eller

eventuelt begge ord forekommer. Vægte på 1 skal bruges, hvis brugeren mener, at des flere af forespørgslens ord, der forekommer i et dokument, des bedre. Ved en vægt på 1 er OG-operatoren og ELLER-operatoren identiske. I begge tilfælde betragtes et dokument, der både indeholder ordet før operatoren og det efter, som dobbelt så godt som et dokument, hvor kun det ene ord forekommer. Ved at bruge vægte mellem 1 og ∞ opnåes en blanding af boolsk søgning og term-vægtning. En ELLER-operator med fx vægt 3 angiver således, at enten det ene eller det andet, men helst begge ord skal forekomme i de dokumenter, der findes frem.

Når søgesystemet skal udføre en udvidet boolsk søgning, foregår det i tre trin: Først udfører systemet en almindelig boolsk søgning. Denne søgning er baseret på en bred forespørgsel, der fremkommer ud fra den udvidede boolske forespørgsel ved at fjerne alle vægtene og erstatte OG-operatorerne med ELLER-operatorer. Dernæst gøres den udvidede boolske forespørgsel komplet ved at tilføje standardværdier for de vægte, brugeren har udeladt. Endelig udføres denne nu komplette udvidede boolske forespørgsel på den delmængde af dokumenterne, der blev fundet frem med den brede forespørgsel. For hvert dokument i denne delmængde beregnes forespørgsel/dokumentligheden ved hjælp af udtrykkene ovenfor, og de fremfundne dokumenter ordnes efter aftagende lighed.

Skimming

Ved boolsk og udvidet boolsk søgning skal den utilstrækkelige viden formuleres i en forespørgsel. Det volder de fleste brugere problemer. En væsentlig grund til disse problemer er, at forespørgslen i første omgang skal formuleres på et tidspunkt, hvor brugeren endnu ikke har set det første eksempel på et relevant dokument. Ved senere reformuleringer af forespørgslen spiller ord fra de allerede fundne relevante dokumenter en stor rolle. Skimming ('browsing') er udtryk for en helt anden måde at anskue søgeprocessen på. Skimming drager fordel af to alment menneskelige evner (Thompson & Croft 1989): For det første har mennesker væsentligt lettere ved at genkende det, de søger, end ved at beskrive det. For det andet er mennesker gode til at 'læse ned over en side' og slå ned på det væsentlige, hvis der er noget.

Disse to evner udnyttes ved at lade brugeren kigge direkte i dokumenterne og ved at gøre det let at følge de associationer, der opstår under læsningen. Det er selvfølgelig kun en udvalgt og meget begrænset mængde associationer, systemet kan støtte. Thompson & Croft (1989) nævner en række eksempler i forbindelse med bibliografiske søgninger:

- Hvis dette dokument er interessant: Hvad andet har dets forfatter(e) skrevet?
- Hvis dette dokument er interessant: Er nogen af dets henvisninger interessante?
- Hvis dette dokument er interessant: Er nogen af de dokumenter, der henviser til dette dokument, interessante?
- Hvis dette dokument er interessant: Hvilke andre dokumenter er der i samme konference-rapport, tidsskriftsnummer osv?
- Hvis dette dokument er interessant: Er der nogle dokumenter i systemets database, der ligner dette dokument meget?
- Hvis dette begreb er interessant: Har det nogle synonymer?
- Hvis dette begreb er interessant: Hvilke dokumenter forekommer det i?

For at muliggøre skimming organiseres søgesystemets dokumentdatabase som et netværk. Dokumenterne er netværkets knuder, og associationerne - de kaldes normalt referencerne - udgør forbindelserne mellem knuderne. Når brugeren under læsningen af et dokument finder fx en interessant henvisning, markeres denne - det sker typisk ved hjælp af en mus. Efter at markeringen er foretaget, finder systemer det dokument, der henvises til, frem.

Et væsentligt problem i forbindelse med skimming er, at det er let at fare vild. Det er svært at bevare overblikket, når søgningen foregår ved at hoppe fra et dokument til et andet. Dette problem søges reduceret ved hjælp af overblikskort (Thorup m.fl. 1990). Et overblikskort kan enten være statisk og vise de referencer, der udgår fra et givet dokument, eller det kan være dynamisk og vise hvilke dokumenter og referencer, brugeren har fulgt i løbet af søgningen.

Skimming betragtes ikke som et alternativ til (udvidet) boolsk søgning, men som et værdifuldt supplement (Palay & Fox 1981), (Thompson & Croft 1989). Skimming er velegnet til nogle søgninger, men i mange tilfælde savnes den fuldstændighed i søgningerne, som boolsk og udvidet boolsk søgning er bygget op om. Thompson & Croft (1989) foreslår, at brugeren først bruger skimming til at afklare og præcisere deres informationsbehov og derefter fx boolsk søgning.

2.5 Faciliteter til at forbedre søgningen

I dette afsnit beskrives fire faciliteter, der ikke i sig selv er søgeteknikker, men kan bidrage til at forbedre søgninge n. Det drejer sig for det første om opdeling af databasen i klynger, for det andet om tesaurusser, for det tredje om joker-operatorer og automatisk behandling af bøjninger og for det fjerde om reformulering af forespørgsler og relevans-feedback.

Opdeling af databasen i klynger

Opdeling af fuldtekstdatabaser i klynger er dels baseret på indekserings- dels på klynge-teknikkerne ('clustering'), der begge er veletablerede indenfor informationssøgning generelt. Kombinationen af fuldtekstsøgning og klynger er imidlertid på et eksperimenterende stadium (Tenopir 1984).

Formålet med opdeling af databasen i klynger er at give brugeren mulighed for at begrænse søgningerne til en del af databasen. Opdelingen skal altså gruppere dokumenterne i en række klynger, der hver omfatter et (bredt) emne. En sådan opdeling vil ofte være et hierarki, hvor hvert dokument placeres i én klynge; men det behøver ikke være tilfældet. Når det ikke er tilfældet, bliver opdelingen i klynger en slags grov indeksering, hvor placeringen af et dokument i en klynge svarer til at hæfte et nøgleord på dokumentet. Sprowl (1981) anbefaler kombinationen af fuldtekstsøgning og en sådan grov indeksering fremfor fuldtekstsøgning alene.

Fordelen ved at opdele databasen i klynger er, at søgningen omfatter færre dokumenter. Det har to konsekvenser: Det betyder for det første, at færre irrelevante dokumenter ved et tilfælde matcher forespørgslen. Derfor kan søgningen gøres bredere - for at få et højt *recall* - uden, at *precision* bliver urimeligt lav. For det andet går søgningen hurtigere. Det vil sige, at der kan stilles mere komplicerede forespørgsler, hvis de kan begrænses til en del af databasen. Ulempen ved at afgrænse søgningen til en del af databasen er, at nogle af de relevante dokumenter ofte vil ligge udenfor de klynger, søgningen foretages i.

I forbindelse med opdelingen af databasen i klynger er det relevant at skelne mellem to typer fuldtekstsøgesystemer afhængig af, om dokumenterne udgør et samlet hele, eller der er tale om en samling uafhængige dokumenter. Som eksempel på det første kan nævnes et juridisk informationssøgesystem, hvor dokumenterne udgøres af love, bekendtgørelser og cirkulærer. Her udgør dokumenterne på sin vis én samlet tekst, der blot er struktureret i en række dokumenter. De enkelte dokumenter er skrevet under hensyntagen til, hvad der står i de øvrige, og der er få gentagelser. Et eksempel på den anden type søgesystem er et juridisk informationssøgesystem, hvor hvert dokument svarer

til en sag. Her er hvert dokument selvstændigt i forhold til de øvrige, og lovgrundlaget inddrages i utallige forskellige kombinationer. Der er følgelig mange gentagelser, og det er svært at fastlægge en struktur. Den første type søgesystemer er meget mere velegnet til opdeling i klynger, idet teksten allerede har en struktur, behandlingen af hvert emne er søgt samlet ét sted, og der er få gentagelser. Ydermere er tekstens struktur ofte allerede gjort eksplicit gennem en indholdsfortegnelse.

Tesaurusser

Ordet 'tesaurus' kommer fra græsk ('thesauros') og latin ('thesaurus'), hvor det betyder skat og skatkammer; men det har i flere hundrede år også haft betydningen ordbog/begrebsordbog. En tesaurus er en vigtig facilitet i forbindelse med fuldtekstsøgning, da den tilbyder brugeren støtte i valget af de ord, forespørgslen skal indeholde. Det er imidlertid kun få fuldtekstsøgesystemer, der har en on-line tesaurus (Tenopir & Ro 1990). Ordet 'tesaurus' er blevet brugt i flere varierende betydninger (Foskett 1985). I det følgende bruges det i overensstemmelse med ISO's definition (ISO 1974):

> *A thesaurus may be defined either in terms of its function or its structure.*
>
> *In terms of function, a thesaurus is a terminological control device used in translating from the natural language of documents, indexers and users into a more constrained "system language" (documentation language, information language).*
>
> *In terms of structure, a thesaurus is a controlled and dynamic vocabulary of semantically and generically related terms which covers a specific domain of knowledge.*

Tesaurusser adskiller sig fra ordbøger på to måder: For det første er en tesaurus organiseret ud fra den indholdsmæssige sammenhæng mellem begreberne, mens en ordbog er organiseret alfabetisk. For det andet beskriver en tesaurus begreberne ved deres relationer til andre begreber, mens en ordbog giver definitioner af begreberne. En meget omfattende liste over tesaurus-relationer findes i (Soergel 1974); de mest udbredte er:

- BT Reference til et bredere begreb i et begrebshierarki. BT-relationen kan kun referere til ét bredere begreb; men der findes andre relationer for referencer til bredere begreber, så tesaurussen kan omfatte flere hierarkier, fx mængde/delmængde og helhed/del.
- NT Reference til snævrere begreber i et begrebshierarki; modsat BT.
- RT Reference til relaterede termer. Ex: *se også* skimming.
- ST Reference til synonyme termer.
- USE Reference til den foretrukne term i en gruppe af synonymer. Ex: *se* fuldtekstsøgesystem.
- UF Reference til termer, som denne term foretrækkes for; modsat USE. Ex: *bruges for* fritekstsøgesystem.
- SN ('Scope Note') Præcisering af begrebets anvendelse, af hvilken af et homografs betydninger der er tale om og lignende. Batty (1989) foreslår en speciel type SN:
- HN ('Historical Note') Til angivelse af ændringer i en terms brug eller form, så den tidligere brug/form forbliver i systemet og stadig kan forstås.

SN og HN er faktisk ikke relationer; de er noter til det ene begreb, de er knyttet til. SN giver således mulighed for at knytte oplysninger til de enkelte begreber i tesaurussen. Buchan (1989) foreslår, at tesaurusser og ordbøger sammenskrives ved, at SN udvides

med en definition af begrebet. Nedenfor følger et simpelt - og ufuldstændigt - eksempel på, hvordan en indgang i en tesaurus kan se ud:

Fuldtekstsøgesystem
BT informationssøgesystem
NT LEXIS
RT invertering, stopliste, skimming
ST fritekstsøgesystem
UF fritekstsøgesystem
SN den type informationssøgesystemer, hvor dokumenternes fulde tekst er til rådighed i systemet, og søgning foregår i hele teksten.

Tesaurusser har en meget vigtig funktion i forbindelse med fuldtekstsøgning: De medvirker til at reducere problemet med, at brugeren med sine søgeord skal ramme netop de ord, der forekommer i de relevante dokumenter. Tesaurusser kan bruges til at øge *recall* ved automatisk at udvide forespørgslen, så søgningen også omfatter fx bredere og synonyme begreber. Ved at erstatte et eller flere søgeord med snævrere begreber kan tesaurusser ligeledes bruges til at øge *precision*. Foskett (1985) angiver, at tesaurusser har til formål:

- At støtte brugerne af et søgesystem i valget af søgeord.
- At udgøre et hierarki, der kan bruges til at udvide og indsnævre søgninger - ved at vælge bredere/snævrere søgeord.
- At udgøre et 'kort' over et givet fagområde, ved at angive hvordan begreberne er relateret til hinanden.
- At give mulighed for at indplacere/lokalisere nye begreber og deres relationer til det allerede eksisterende netværk af begreber på en overskuelig måde.
- At udgøre et standardiseret/kontrolleret glosarium og derved støtte en standardisering af terminologien indenfor et fagområde - blandt andet ved hjælp af USE-referencer.

Tesaurusser skal naturligvis vedligeholdes. I så godt som alle tilfælde beskrives det som en proces, der kræver central styring, se fx (Soergel 1974), (Strong & Drott 1986) og (Batty 1989). Over en periode indsamles forslag til ændringer og nye ord, derefter overlades det til en lille gruppe at ajourføre relationerne og indplacere de nye ord. Den centrale styring anses for nødvendig for at sikre en konsistent tesaurus.

Vi har fundet en enkelt artikel (Güntzer m.fl. 1989), der handler om muligheden for at gøre vedligeholdelsen til en brugerstyret, semiautomatisk proces: Tesaurussen indgår i et informationssøgesystem, TEGEN, der bruges hyppigt af flere personer. I forbindelse med forespørgslerne kan brugerne komme med forslag til nye ord og relationer mellem ord; systemet gemmer forslagene med en markering af, at de er ubekræftede. Når en anden bruger senere slår op i tesaurussen, vil systemet vise både de allerede fastlagte ord og relationer og - med en tydelig markering - de ubekræftede.

Brugeren bliver bedt om at under- eller anerkende de ubekræftede forslag, og disse vurderinger tælles op over en periode. Afhængigt af hvor positivt gruppen af brugere har vurderet et forslag, bliver det enten forkastet, accepteret eller fortsætter som ubekræftet i næste periode. Da der er tale om et forskningsprojekt, udføres sideløbende en grundig manuel kontrol. Intentionen er imidlertid, at det manuelle arbejde kan reduceres til stikprøvekontroller samt løbende vurderinger af de forslag, der er relevante for brugernes øjeblikkelige forespørgsler.

Omfattende udvidelser, fx inddragelse af et nyt emneområde, er et andet aspekt af vedligeholdelsen af tesaurusser. En mulighed i denne retning er at udvide en tesaurus ved automatisk sammenfletning af tesaurusser for forskellige områder (Rada & Martin 1987),

(Mili & Rada 1988).

Hvis en tesaurus skal fungere, må den nødvendigvis ændre sig med ændringerne i sprogbrug. Hvadenten disse ændringer samles i portioner eller foretages løbende, stiller det krav om en fleksibel datastruktur. Flere datastrukturer, såvel enkle som mere avancerede, forekommer relevante i forbindelse med tesaurusser. Relationsdatabaser er velegnede og bruges ofte; to andre muligheder er objekter i objekt-orienteret programmering (Kleinbart 1985) og semantiske net (Mili & Rada 1988), (Hoppe m.fl. 1990).

Joker-operatorer og automatisk behandling af bøjninger

Et centralt problem i forbindelse med fuldtekstsøgning er forskellen på begreber og termer (pånær i dette afsnit bruger vi de to ord uden at skelne). Begreber er abstrakte og bruges til at organisere menneskers viden; termer er ord/data og bruges til at betegne begreber. Sammenhængen mellem begreber og termer fastlægges af terminologien. Tesaurusser er ét forsøg på at lette springet fra brugerens begrebsverden til dokumenternes termer. Udover springet fra begreb til term har termerne i sig selv nogle egenskaber, der komplicerer søgning. De egenskaber, der giver anledning til problemer, omfatter blandt andet:

- Homografer, ord der skrives ens, men iøvrigt er forskellige.
- Sammensatte ord, fx 'relationsdatabase' (det er ofte ønskeligt at kunne søge på ordets enkelte dele).
- Bøjninger, fx ental/flertal.
- Varierende stavemåder, fx 'ressource' og 'resurse'.
- Store/små bogstaver, fx EDB = edb, men ROM ≠ Rom ≠ rom.
- Forkortelser, fx DB for database.

Det mest udbredte tiltag til at løse flere af disse problemer er at give brugeren mulighed for ved hjælp af joker-operatorer ('wildcards') at søge på dele af ord. Det gøres typisk ved at tillade to specielle tegn, fx '?' og '*', i forespørgslerne. Det første kan stå i stedet for ét vilkårligt bogstav, det andet for en gruppe på ingen, ét eller flere vilkårlige bogstaver. Der er to oplagte problemer ved dette: For det første er den faste del af et søgeord ofte så lille, at den ikke kan bruges. Ex: 'ressource' kan ikke erstattes af 'res*'. For det andet kan en alfabetisk ordnet indeksfil ikke behandle søgeord som '*database' effektivt.

Af de ovennævnte problemer er problemet omkring bøjninger blevet ofret særlig opmærksomhed. Hvis brugeren er interesseret i begrebet 'indeksering', bør dokumenter, hvor fx ordet 'indekseringen', 'indekseringernes' eller det tilsvarende verbum 'indeksere' forekommer, også findes frem. Joker-operatorer kan bruges til dette formål; afskæring er en anden mulighed. Ved afskæring søges ordene reduceret til deres grundformer ved afskæring af endelser. I enkelte tilfælde er afskæring af forstavelser også mulig. Afskæring kan ske ved hjælp af en tabel over standard-endelser og en antagelse om, at hvis et ords endelse findes i denne tabel, så fremkommer grundformen ved at fjerne denne endelse fra ordet (Salton & McGill 1983). En sådan kontekstfri regel giver selvfølgelig nogle problemer: Hvis 'else' er i endelsestabellen, afskæres 'tilføjelse' korrekt til 'tilføj'. Ordet 'helse' afskæres derimod fejlagtigt; problemer som dette løses ved at fastsætte en minimumslængde for grundformen, typisk 3 tegn (van Rijsbergen 1979). Ordet 'værelse' afskæres imidlertid stadig fejlagtigt og endda til stammen af verbet 'være'. Denne type utilsigtede sammenfald er begrundelsen for den afskæringsteknik, der omtales nedenfor.

En mere avanceret mulighed er at basere afskæringen på en tabel over ækvivalente endelser (van Rijsbergen 1979). For at afskære to ord til den samme stamme skal ordene være ens pånær deres endelser, og endelserne skal være ækvivalente. Ex: Ordene

'interesse' og 'interessant' afskæres til samme grundform, hvis 'se' og 'sant' er i tabellen over ækvivalente endelser.

Joker-operatorer og faciliteter til automatisk behandling af bøjninger kan ligesom tesaurusser bruges til at udvide og indsnævre søgningerne. Normalt vil brugeren være interesseret i, at søgningen udvides ved automatisk tilføjelse af fx søgeordenes forskellige bøjninger. I nogle tilfælde er det imidlertid ønskeligt at kunne indsnævre søgningen til fx en bestemt kombination af store og små bogstaver. En søgning efter byen Rom med søgeordet 'Rom' (med stort 'R') vil give højere *precision* end en søgning, hvor der ikke skelnes mellem store og små bogstaver. I næste afsnit ser vi nærmere på, hvordan muligheden for at udvide og indsnævre søgningen kan udnyttes.

Reformulering af forespørgsler og relevans-feedback

Interaktive fuldtekstsøgesystemer giver mulighed for, at systemet kan støtte brugeren i formuleringen af den utilstrækkelige viden. Det ligger lige for at organisere arbejdet med formuleringen af forespørgslen som en iterativ proces, dvs at reformulere forespørgslen ud fra en hurtig, overfladisk vurdering af de dokumenter, den forrige forespørgsel resulterede i. Reformuleringen af forespørgslen kan være helt overladt til brugeren; i andre systemer foregår reformuleringen automatisk, brugeren skal blot give relevans-feedback.

Relevans-feedback består i en angivelse af, om der er fundet for få eller for mange dokumenter frem, eventuelt kombineret med en angivelse af hvilke af de fremfundne dokumenter, der er relevante. Systemet bruger disse oplysninger til automatisk at udvide eller indsnævre søgningen og - i tilfælde af udvidet boolsk søgning - justere søgeordenes vægte. Ingwersen (1984) har foreslået, at den automatisk reformulerede forespørgsel forelægges for brugeren. Det giver brugeren mulighed for at ændre i forespørgslen, før den udføres; den kan naturligvis også accepteres uændret.

Gauch & Smith (1989) beskriver et søgesystem, der kan reformulere forespørgsler ud fra relevans-feedback. Denne relevans-feedback består blot i en angivelse af, om søgningen har resulteret i for få eller for mange dokumenter. I reformuleringen deles ordene i forespørgslen i to grupper: De inkluderede og de ekskluderede. De inkluderede ord er de ord, forespørgslen angiver, at dokumentet skal indeholde. De ekskluderede ord er de ord, der omsluttes af en IKKE-operator. En søgning udvides ved at slække kravene til de inkluderede ord og stramme kravene til de ekskluderede; omvendt når en søgning indsnævres. Der er flere forskellige muligheder for at slække/stramme kravene. Gauch & Smith har fastlagt en bestemt rækkefølge, disse muligheder benyttes i.

Når en søgning udvides, sker der følgende med de inkluderede ord: Først tilføjes ordenes grundformer til forespørgslen, så tilføjes synonymer, og derefter slækkes nærhedsoperatorerne (fx fra samme sætning til plus/minus én sætning). Hvis det ikke er tilstrækkeligt, tilføjes først bredere, så relaterede og derefter snævrere begreber til forespørgslen. Hvis forespørgslen stadig giver for få dokumenter, slækkes nærhedsoperatorerne yderligere (fx fra samme sætning til samme afsnit), så erstattes OG-operatorerne med ELLER-operatorer, og endelig fjernes IKKE-operatorerne. De ekskluderede ord behandles omvendt, som det fremgår af nedenstående oversigt:

Udvidelse af forespørgslen	Inkluderede ord	Ekskluderede ord
Grundformer	tilføjes	fjernes
Synonymer	tilføjes	fjernes
Nærhedsoperatorer	slækkes	strammes
Bredere begreber	tilføjes	fjernes
Relaterede begreber	tilføjes	fjernes
Snævrere begreber	tilføjes	fjernes
Nærhedsoperatorer	slækkes yderligere	strammes yderligere
Boolske operatorer	slækkes	strammes
IKKE-operatorer	fjernes	

Figur 2.8. Oversigt over teknikker til udvidelse af forespørgsler, efter (Gauch & Smith 1989). Inkluderede ord er de ord, som forespørgslen angiver, dokumenterne skal indeholde. Ekskluderede ord er de ord, der indgår som IKKE(ord).

Det er afgørende, at relevans-feedback kan gives ud fra en meget hurtig og overfladisk vurdering af de fremfundne dokumenter. Hvis brugerne ikke kan nøjes med at læse resuméet eller lignende, men skal til at skimme selve teksten, så er de formodentlig langt bedre i stand til at reformulere forespørgslen selv (van Rijsbergen 1979).

2.6 Brugergrænsefladen

Med de interaktive søgesystemer er brugergrænsefladen blevet en central problemstilling. Interessen for brugergrænsefladen betyder, at der nu eksperimenteres med fuldtekstsøgesystemer, hvor brugergrænsefladen er grafisk og baseret på direkte manipulation. Som eksempel bruger McMath m.fl. (1989) en grafisk brugergrænseflade med mus til at vise tesaurussen og vælge ord fra den. Der bruges også ekspertsystemteknikker til at opbygge brugermodeller. Ved hjælp af en sådan model kan interaktionen mellem søgesystem og bruger afhænge af, om brugeren er novice eller ekspert, om brugeren ønsker overblik eller specifikke detaljer osv. Det betyder, at den hjælp, der gives on-line, såvel som søgningernes bredde/dybde osv kan tilpasses den enkelte bruger (Thompson & Croft 1989).

Vi afgrænser dette afsnit til en kort beskrivelse af to aspekter ved brugergrænsefladen. Baggrunden for denne afgrænsning er, at det ikke er søgesystemernes brugergrænseflade, men deres funktionalitet og fleksibilitet, der er i fokus i dette speciale. Begge de behandlede aspekter er specifikt knyttet til søgesystemer. Det drejer sig om: Forespørgsler som formelle udtryk eller i naturligt sprog og præsentationen af de fremfundne dokumenter.

Forespørgsler som formelle udtryk eller i naturligt sprog

I såvel boolsk søgning som udvidet boolsk søgning er forespørgslerne formelle udtryk. Det muliggør meget specifikke forespørgsler, men forudsætter, at brugerne er fortrolige

med de boolske operatorer. Forespørgsler i naturligt sprog vil reducere problemet med at formulere den utilstrækkelige viden, da en almindelig, sammenhængende, skriftlig beskrivelse af det, brugeren vil vide noget om, kan bruges som forespørgsel.

Udvidet boolsk søgning giver mulighed for forespørgsler i naturligt sprog. Brugeren formulerer forespørgslen som en almindelig tekst; der er ikke nogen begrænsninger i sætningsopbygning, ordvalg eller lignende. Derefter transformerer søgesystemet denne tekst til en udvidet boolsk forespørgsel, som søgningen derefter baseres på. Genereringen af den udvidede boolske forespørgsel er et eksempel på automatisk indeksering: En tekst i naturligt sprog repræsenteres ved en række vægtede ord. Genereringen af de udvidede boolske forespørgsler foregår således ved hjælp af teknikker til automatisk indeksering: De ord, der hverken har særlig lav eller særlig høj frekvens, betragtes som gode indikationer af forespørgslens indhold og bruges i den udvidede boolske forespørgsel. En detaljeret gennemgang af automatisk indeksering findes fx i (Salton & McGill 1983), (Salton 1986) og (Salton & Buckley 1988).

Præsentationen af de fremfundne dokumenter

En forespørgsel fører til, at en mængde af dokumenter findes frem. Disse dokumenter skal brugeren præsenteres for. Det kan ske i form af en oversigt med de pågældende dokumenters titler og lignende oplysninger. Ud fra denne oversigt er det op til brugeren at foretage vurderingen af dokumenterne. Denne mulighed er meget simpel og kan forbedres på flere måder.

En væsentlig mulighed for at forbedre præsentationen er passage-præsentation (O'Connor 1975). Ved passage-præsentation vises for hver forekomst af et af forespørgslens søgeord den passage, ordet forekommer i. En passage kan være en linie, en sætning eller lignende. I enkelte tilfælde kan passagernes størrelse varieres, så brugeren selv kan definere hvor meget tekst, der skal vises på hver side af søgeordet. Fordelen ved passage-præsentation er, at søgeordenes kontekst inddrages i præsentationen.

Passager giver brugeren væsentligt bedre muligheder for at vurdere et dokuments relevans end bibliografiske oplysninger - fx forfatter, titel og årstal - alene. Passager giver dels brugeren bedre mulighed for at vurdere selve dokumentet, dels giver de bedre mulighed for at vurdere, hvorfor søgesystemet fandt dette dokument frem. Passage-præsentation giver gode muligheder for umiddelbart at sortere mange af de irrelevante dokumenter fra. Det er fx tilfældet, når søgeordet 'fuldtekstsøgning' giver passagen "Vi vil således ikke komme ind på fuldtekstsøgning", eller når søgeordet 'hånd' giver en passage med vendingen "på egen hånd".

Hvis databasen er opdelt i klynger, er en anden mulighed for at forbedre præsentationen at angive den klynge (eller de klynger), dokumentet tilhører (Lesk 1989). Denne oplysning er mest værdifuld, hvis hvert dokument kun behandler ét emne eller opdelingen tillader, at et dokument tilhører flere klynger. Ellers fortæller klyngerne kun noget om hvert enkelt dokuments hovedemne, mens dokumenterne også indeholder noget relevant om sekundære emner.

2.7 Sammenfatning

I dette kapitel har vi givet en oversigt over *state of the art* for fuldtekstsøgesystemer. Vi har behandlet historie, lagringsteknikker, tekstrepræsentation, søgeteknikker, faciliteter til at forbedre søgningen og kort omtalt visse aspekter ved brugergrænsefladen.

Vi har karakteriseret hovedstrømningerne indenfor forskningen i informationssøgesystemer ved hjælp af tre paradigmer - det system-drevne, det bruger-orienterede og det kognitive paradigme. I øjeblikket er det kognitive paradigme ved at

etablere sig. Med det kognitive paradigme tænkes og udvikles søgesystemerne i en sammenhæng, der omfatter såvel de tekniske aspekter som brugssituationen. Det er efter vores mening helt nødvendigt at betragte de tekniske aspekter og brugssituationen i en snæver sammenhæng. Denne erkendelse er imidlertid ikke opstået med det kognitive paradigme, selvom der her lægges stor vægt på den. Derudover markerer det kognitive paradigme en interesse for nogle af de teknikker, der er udviklet indenfor forskningen i kunstig intelligens. Vi ser ingen muligheder for at udvikle intelligente systemer. Fuldtekstsøgesystemer såvel som andre edb-systemer skal efter vores mening fungere som redskaber for brugerne - der skal være tale om maskinstøttet intelligens.

Fuldtekstsøgning og nøgleordsbaseret søgning sammenlignes ofte og betragtes tit som hinandens modsætninger. Sammenligningerne påvirkes af mange parametre, fx af om brugerne har en tesaurus til rådighed ved fuldtekstsøgning, og har da også givet noget forskellige resultater. Generelt siges fuldtekstsøgning at give højere *recall* end nøgleordsbaseret søgning, mens nøgleordsbaseret søgning giver højere *precision*. Det betyder, at fuldtekst-søgning er at foretrække i nogle situationer, mens nøgleordsbaseret søgning er det i andre. Der er således behov for søgesystemer, der tilbyder begge muligheder (Tenopir 1984).

Situationen er den samme for søgeteknikkerne. Belkin & Croft (1987) pointerer, at én søgeteknik ikke er tilstrækkeligt til at dække alle behov. I de kommercielt tilgængelige systemer er boolsk søgning med nærhedsoperatorer næsten enerådende. Boolsk søgning udsættes for megen kritik og søges i konsekvens af det ofte ændret eller udvidet. Belkin & Croft ser et behov for at kombinere boolsk søgning, der er en eksakt match teknik, med en partiel match teknik; et eksempel på dette er udvidet boolsk søgning. Begrebsbaseret fremfinding er et andet forsøg på at forbedre boolsk søgning. Begrebsbaseret fremfinding er rettet mod brugergrænsefladen, der er udformet så den støtter brugeren ved at bidrage til at strukturere den utilstrækkelige viden.

Ved fuldtekstsøgning er der endvidere behov for faciliteter, der kan bidrage til at reducere springet fra de begreber, brugeren tænker i, til de termer, der forekommer i dokumenterne. De to mest udbredte faciliteter til dette er joker-operatorer og tesaurusser. Et helt andet tiltag med samme sigte er bestræbelserne på at give brugeren mulighed for at stille sine forespørgsler i naturligt sprog.

I dette kapitel har vi beskrevet de væsentligste elementer i fuldtekstsøgesystemer, som de behandles i litteraturen. Denne beskrivelse udgør den tekniske del af grundlaget for den følgende case, hvor vi designer og konstruerer en prototype på et fuldtekstsøgesystem. I relation til vores idé om at basere fuldtekstsøgesystemer på relationsdatabaser har vi beskrevet de lagringsteknikker, der benyttes i forbindelse med fuldtekstsøgesystemer, og den kritik, der rettes mod brugen af relationsdatabaser. Relationsdatabaser kritiseres først og fremmest for at bruge for meget lagerplads og resultere i for lange svartider. Vi vil efterprøve denne kritik nærmere, da vi mener, at relationsdatabasernes funktionalitet og fleksibilitet mere end opvejer ulemperne. Relationsdatabasernes funktionalitet og fleksibilitet gør dem efter vores mening velegnede som grundlag for systemer, der kan tilpasses de løbende ændringer i brugerens arbejdssituation.

Mens dette kapitel udgør den tekniske del af grundlaget for vores case, vil vi i det næste kapitel vende os mod den brugssituation, vores case udspiller sig i.

3. Juridisk sagsbehandling - juridiske informationssøgesystemer

Efter gennemgangen af *state of the art* for fuldtekstsøgesystemer vil vi nu vende os mod en case. Formålet med denne case er først og fremmest at illustrere og afprøve vores ideer om fuldtekstsøgesystemer, der kan udvikle sig over tid. I den forbindelse får vi også mulighed for at illustrere og vurdere en række af de teknikker og forslag, der er beskrevet i forrige kapitel.

I dette kapitel introduceres det anvendelsesområde for fuldtekstsøgesystemer, vi vil beskæftige os med i de følgende kapitler: Juridisk sagsbehandling. Vi forsøger at indkredse juristers arbejde og de krav, de stiller til et juridisk informationssøgesystem. I kapitel 4 udvikler vores case sig til en prototype: Edb-Karnov. Vi diskuterer, fastlægger og implementerer udvalgte dele af et juridisk fuldtekstsøgesystem baseret på Karnovs Lovsamling. I kapitel 5 kommer vi med et forslag til, hvordan de ændringer, brugeren foretager løbende, kan integreres med udgivelsen af nye udgaver af Edb-Karnov. Denne integration er et afgørende aspekt i Edb-Karnovs udvikling over tid, da faciliteterne til udvikling over tid ikke vil blive brugt, hvis ændringerne går tabt med udgivelsen af en ajourført udgave af Edb-Karnov. Endelig afslutter vi vores case i kapitel 6 med en demonstration af Edb-Karnov.

Juridiske informationssøgesystemer har længe været det førende område indenfor fuldtekstsøgesystemer (Tenopir 1984). Der er tre grunde til, at vi vælger juridisk sagsbehandling og juridiske informationssøgesystemer som emne for vores case. For det første har Erik Frøkjær, der gav os ideen til specialet og etablerede kontakten til Karnovs Forlag, udviklet den i forbindelse med juridisk præget sagsbehandling. For det andet vil vi beskæftige os med søgesystemer, der skal fungere som redskaber for fagfolk. Fagfolks situation og råderum i de organisationer, hvor de er ansat, er anderledes, end det er tilfældet for personer med mere rutineprægede arbejdsopgaver. De har derfor mulighed for at stille andre og større krav til deres redskaber. Jurister er et godt eksempel på fagfolk, og de stiller, som det vil fremgå af dette kapitel, store krav til juridiske informationssøgesystemer. Den tredje grund til, at vi vælger dette anvendelsesområde som emne for vores case, er muligheden for at basere vores case på et meget relevant datamateriale. Denne mulighed skylder vi Karnovs Forlag, der har stillet godt 400 sider af Virksomheds-Karnov til rådighed for os på maskinlæsbar form.

Da vi begyndte på projektet, havde vi kun et meget overfladisk kendskab til jura, jurister, deres arbejde og de hjælpemidler, de har til rådighed. Vi er gået flere veje i vores bestræbelser på at erhverve os tilstrækkelig indsigt til at kunne gennemføre vores case: Vi har interviewet to jurister; haft flere samtaler med direktør Jens Peter Nielsen, Karnovs Forlag; været til en af Datacentralens demonstrationer af Retsinformation; læst en del litteratur, fx (Blume 1989) og (Bache 1991), som jurister har skrevet om deres arbejde og behov for edb; og vi har trukket på vores vejleder, Erik Frøkjær, der har flere års erfaring med juridisk præget sagsbehandling. Endelig har vi forladt os på, at vores egne tanker og sunde fornuft ikke ramte helt ved siden af.

I dette kapitel, specielt afsnit 3.1, gør vi rede for de centrale dele af den viden, vi i løbet af projektet har fået om juridisk sagsbehandling og de krav, jurister stiller til søgesystemer. Kapitlet munder ud i en kort beskrivelse af Edb-Karnov, det juridiske fuldtekstsøgesystem, vi vil bruge til at illustrere og afprøve vores ideer om søgesystemer, der kan udvikle sig over tid. Kapitlet består af seks afsnit:

1. Juridisk sagsbehandling

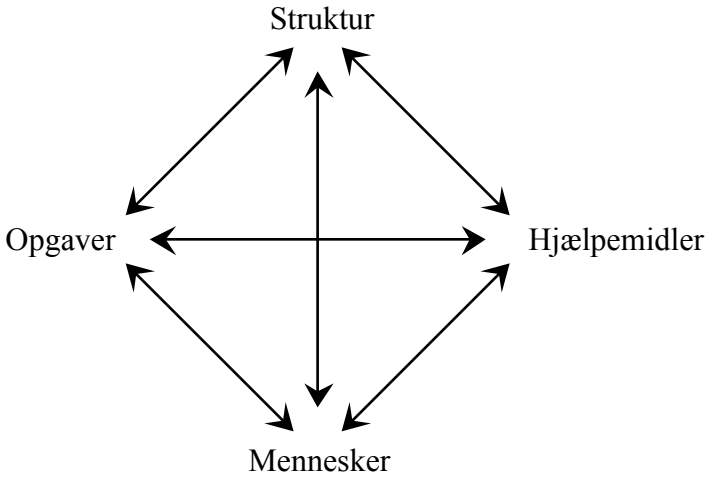
2. Eksisterende hjælpemidler
3. Udvikling over tid
4. Fuldtekstsøgning overfor nøgleordsbaseret søgning
5. Introduktion til Edb-Karnov
6. Sammenfatning

3.1 Juridisk sagsbehandling

I dette afsnit vil vi prøve at indkredse arbejdsområdet juridisk sagsbehandling - hvad består juristers arbejde med en sag i? Hovedkilden til dette afsnit er to interviews om indholdet af juristers arbejde og deres ønsker, behov, visioner og forbehold i forbindelse med brug af edb i juridisk sagsbehandling. De to personer, vi har interviewet, er jurist Dorthe la Cour, Dansk Arbejdsgiverforening, og advokat Per Sjøqvist, advokatfirmaet Horten & Co. For en nærmere beskrivelse af forberedelserne, forløbet og efterbehandlingen af interviewene henvises til kapitel 7. Vi vil imidlertid starte med en kort, overordnet og teoretisk beskrivelse af jurister og de omgivelser, de typisk arbejder i.

Fagfolk i et fagbureaukrati

Leavitt (1964) har i en klassisk artikel lokaliseret fire centrale variable i beskrivelsen af enhver organisation: Strukturen, menneskene, opgaverne og hjælpemidlerne i form af teknologi og andre redskaber (se figur 3.1). De fire variable påvirker alle hinanden. Ændringer i én af de variable giver også anledning til ændringer i de øvrige; ligesom en ny forståelse af én variabel også nødvendiggør nytænkning af de øvrige. Vi vil først indkredse jurister og deres arbejde med udgangspunkt i strukturen, derefter vender vi os mod de tre andre variable.



Figur 3.1. Leavitts organisationsmodel (Leavitt 1964). Modellen udpeger fire centrale variable i beskrivelsen af enhver organisation og understreger, at de udgør en helhed, der kun kan forståes i sammenhæng.

Et meget udbredt udgangspunkt for studier af organisationer er strukturen. En anerkendt syntese af begreberne og resultaterne indenfor denne tilgang til organisationsteorien er (Mintzberg 1983). Mintzberg (1983) bruger advokatkontoret og universitetet som typeeksempler på den organisationstype, han kalder fagbureaukratiet. Hver organisationstype - der er fem ialt - kendetegnes ved en dominerende koordinationsmekanisme og den centrale medarbejdergruppe. I fagbureaukratiet er den

dominerende koordinationsmekanisme standardisering af de ansattes færdigheder (i modsætning til fx direkte overvågning eller standardisering af arbejdsprocessen). Den centrale medarbejdergruppe i fagbureaukratiet er de menige ansatte (i modsætning til fx ledelsen eller mellemlederne).

I fagbureaukratiet er den vertikale arbejdsdeling meget begrænset; de menige ansattes opgaver opfattes som organisationens primære aktiviteter, mens ledelse og lignende betragtes som støttefunktioner. Den horisontale arbejdsdeling er derimod udtalt; hver enkelt ansat er specialiseret indenfor et snævert område. Der er således kun få, der er i stand til at give en faglig vurdering af en anden ansats arbejde, og der lægges kun lidt vægt på at kontrollere de ansattes indsats. Den vurdering og kontrol, der skal sikre kvaliteten og koordineringen af arbejdet, er i vid udstrækning overladt til den enkelte ansatte selv.

Der foregår selvfølgelig en vis central koordination; men det sker typisk på et overordnet plan, ikke ved direkte indblanding i de menige ansattes daglige arbejde. Det væsentligste element i den centrale koordination er kontrollen med ansættelser og fyringer. Ved ansættelserne består koordinationen primært i at vælge de ansatte fra en lille, ensartet gruppe. Organisationen forlader sig således på den skoling og faglige kompetence, de ansatte fik under deres uddannelse. Den dominerende koordinationsmekanisme er kun at ansætte personer med standardiserede færdigheder erhvervet gennem lange, veletablerede uddannelser.

Fagbureaukratiet er ikke velegnet indenfor alle fagområder. Hvis et fagbureaukrati skal fungere og kunne overleve, stiller det nogle krav til fagområdet. Fagområdet skal være komplekst, ellers er fagbureaukratiet ikke effektivt nok til at konkurrere med de andre organisationstyper. Det er fagområdets kompleksitet, der nødvendiggør, at de menige medarbejdere er højtuddannede specialister, og det er ligeledes fagområdets kompleksitet, der gør overvågning og lignende koordinationsmekanismer uegnede. Fagområdet skal imidlertid også være stabilt. Fagbureaukratiet er en meget individualiseret organisationsform, der ikke har nogen udbredt tradition for samarbejde på tværs af de enkelte ansattes specialer og interesser. Derfor er fagbureaukratiet ikke omstillingsdygtigt nok til at konkurrere med de andre organisationstyper indenfor dynamiske fagområder.

Jurister er fagfolk. De er specialiserede indenfor ret snævre områder, men har stor frihed og dispositionsret i udøvelsen af deres færdigheder. Juridisk sagsbehandling er et komplekst, men nogenlunde stabilt fagområde, som det kræver lang tids uddannelse at beherske. Den strukturelle, teoretiske beskrivelse giver en overordnet forståelse af jurister og deres arbejde, men siger meget lidt om indholdet af deres daglige arbejde. Det daglige arbejde kan beskrives som et samspil mellem mennesker, opgaver og hjælpemidler. Naur (1965) understreger samspillet: De forhåndenværende hjælpemidler påvirker personernes opfattelse af opgaverne, og personernes forståelse af opgaverne er bestemmende for deres mening om, hvad der er et velegnet eller ønskeligt hjælpemiddel. Vi vil derfor ikke behandle mennesker, opgaver og hjælpemidler hver for sig, men prøve at beskrive dem i det samspil, der er grundlaget for vores to interviewpersoners udsagn og eksempler.

Den juridiske metode

Juristers arbejdsmetode bærer præg af den hierarkiske måde, lovgivningssystemet er opbygget på. Lovgivningssystemet er først og fremmest opbygget af love, bekendtgørelser og cirkulærer. Lovene udformes og vedtages af Folketinget. Blandt lovene indtager grundloven en særstilling; men den har sjældent nogen direkte betydning som retskilde. De øvrige love er de vigtigste retskilder, da de er det mest autoritative udtryk for den herskende retsopfattelse (Blume 1989). Bekendtgørelser er generelle retsforskrifter, der

udstedes af myndighederne. De har hjemmel i lovene og specificerer forhold, der ikke beskrives i lovene, fx fordi der er tale om en rammelov. Cirkulærer udstedes af myndighederne og er i princippet interne: De regulerer alene myndighedernes adfærd. Spørgsmålet, om hvorvidt et cirkulære har hjemmel i loven, bliver ikke kontrolleret, før cirkulæret træder i kraft. Et cirkulære er således kun udtryk for myndighedernes retsopfattelse, ikke nødvendigvis for gældende ret. Denne opdeling er imidlertid mestendels af principiel karakter; i praksis fastlægges store dele af retsforholdene i cirkulærerne (Blume 1991). Udover love, bekendtgørelser og cirkulærer spiller tidligere afgørelser af en vis principiel betydning en væsentlig rolle i fastlæggelsen af, hvad der er gældende ret.

Lovgivningssystemets opbygning afspejles ifølge Per Sjøqvist i den arbejdsmetode, de kommende jurister lærer på jurastudiet. Den juridiske metode foreskriver, at juristen altid arbejder sig fra oven og ned gennem det hierarki, lovgivningssystemet udgør. Juristen starter normalt i en lov og arbejder sig derfra ned gennem hierarkiet ved hjælp af de henvisninger, der findes undervejs. Noterne i Karnovs Lovsamling er en stor støtte i denne proces, da de indeholder henvisninger til mange af de relevante bekendtgørelser, cirkulærer, afgørelser osv. Den juridiske metode følger lovgivningssystemets opbygning og sikrer dermed, at juristen får de generelle regler før undtagelserne og særbestemmelserne. Metoden bruges naturligvis ikke indenfor velkendte retsområder.

Vi er ikke i stand til at afgøre, i hvilket omfang den juridiske metode snarere foreskriver, hvordan sagsbehandling burde foregå, end beskriver, hvordan den faktisk foregår. Den grundighed og systematik, der er essensen i den juridiske metode, kommer imidlertid tydeligt til udtryk i både Dorthe la Cours og Per Sjøqvists udtalelser. Vi vurderer, at den juridiske metode giver et godt indtryk af, hvordan jurister strukturerer deres læsning af lovteksterne. Selve læsningen af lovteksterne er en fortolkning, se fx (Blume 1989) og (Leith 1986). Fortolkningen af en lovtekst afhænger både af den sammenhæng, lovtekstens enkelte lovregler står i, og af den situation, den pågældende lovtekst påtænkes anvendt i. I fortolkningen gør juristen ofte brug af analogislutninger (Frøkjær & Pedersen 1987). Frøkjær & Pedersen påpeger, at fortolkningen også kræver ræsonnementer baseret på sund fornuft, da lovteksterne er komplekse og fyldt med vagt definerede begreber. Fortolkningen af lovteksterne kræver således i næsten alle tilfælde, at der udøves et juridisk skøn (Bache 1991).

Tre typiske sager

Juridisk sagsbehandling omfatter naturligvis meget forskelligartede sager. Dorthe la Cour peger på to meget forskellige grupper sager, der begge er typiske. Den første gruppe kan ofte klares ved et enkelt opslag; den anden kræver omfattende søgninger. Hun mener, disse to grupper markerer grænserne for, hvordan en sag typisk ser ud. Indenfor den anden gruppe skelner Per Sjøqvist mellem to slags sager: De bagudrettede og de fremadrettede. Bagudrettet sagsbehandling drejer sig om at finde det sted i lovteksten, der giver svar på et givet spørgsmål. Fremadrettet sagsbehandling drejer sig op at opstille dokumenter, fx kontrakter, ud fra reglerne i lovteksten. I det følgende gennemgås et eksempel på hver af de tre typer af sager:

Eksempel 1: Opslag

Skal arbejdsgiveren betale sygeferiepenge til arbejdere, der bliver syge?

Det er faktisk en meget simpel sag: Man slår simpelthen op på ferielovens §13, og der står hvilke betingelser, der skal være opfyldt osv. I dette tilfælde står svaret direkte i loven; det

er ikke nødvendigt at læse bekendtgørelser og lignende. Denne type sager vil det hverken blive lettere eller sværere at klare med et edb-systems hjælp.

Eksempel 2: Bagudrettet sagsbehandling

> Hvad er konsekvenserne af, at man glemmer at fortælle sin arbejdsgiver, at man vil have fædreorlov?

Denne sag stammer ligesom det foregående eksempel fra Dorthe la Cours arbejde; men den er væsentlig mere kompliceret. Man skal for det første vide, at hvis lovteksterne indeholder et svar på spørgsmålet, så findes det i funktionærloven, lov om ligebehandling af mænd og kvinder eller sygedagpengeloven. For det andet finder man ikke noget om konsekvenserne i disse tre love. Der står, at orloven skal varsles, men ikke noget om konsekvenserne af at glemme det. Det tredje punkt er at udvide søgningen til egne og kollegers notater. Det gav heller ikke noget i dette tilfælde. Det fjerde punkt er at søge andre steder og/eller på andre områder. Dorthe la Cour brugte ca en uge på sagen, før hun fandt svaret i et brev: En virksomhed i en konkret sygedagpengesag havde spurgt dagpengeudvalget (ankeinstansen for sygedagpengesager) om konsekvenserne. Sygedagpengeudvalget havde sendt spørgsmålet videre til Arbejdsministeriet. Arbejdsministeriets svar er et brev fra april 1988, og det er det eneste sted, hvor svaret på spørgsmålet findes på tryk.

Denne arbejdsproces er tung og så omfattende, at det nærmer sig detektivarbejde. Dorthe la Cour og Per Sjøqvist er enige om, at juridisk sagsbehandling er mindst lige så meget detektivarbejde, som det er sager, der kan klares direkte ved opslag. Dette eksempel går kun et par år tilbage. Per Sjøqvist beskrev en sag, hvor han havde arbejdet sig 100-150 år tilbage i lovgivningen; men det var lidt ud over det sædvanlige.

Eksempel 3: Fremadrettet sagsbehandling

De to foregående eksempler har drejet sig om at finde det sted i lovteksten, der giver svar på et givet spørgsmål. Fremadrettet sagsbehandling drejer sig derimod om at opstille juridisk bindende dokumenter ud fra reglerne i lovteksterne og tidligere dokumenter af samme type.

Som eksempel nævner Per Sjøqvist udarbejdelse af en kontrakt for en totalentreprise. I opstillingen af en sådan kontrakt, der fastlægger reglerne for det fremtidige samarbejde mellem entreprenøren og køberen, er juristen kreativ og skabende. Den jura, der skal bruges, er først og fremmest de grundlæggende regler, om hvad der gælder mellem to parter, fx regler om mangler og forsinkelser. Opgaven løses ved hjælp af tidligere kontrakter og kontorets notater om lignende sager; selve lovteksterne bruges derimod meget lidt.

Har et edb-system noget at tilbyde i forbindelse med sager som disse? Dorthe la Cour har en vision om et juridisk informationssøgesystem; denne vision minder meget om Memex, som Bush (1945) beskrev for 46 år siden. Et centralt element i visionen er muligheden for at tilføje nye tekster til systemet. Det er usandsynligt, at et brev, som det fra eksempel 2, vil være i systemet i forvejen; men det vil være rart at kunne gemme brevet, nu det er fundet. Så er detektivarbejdet gjort én gang for alle. Derudover kan bagudrettet sagsbehandling involvere søgning i gamle dokumenter, hvor sprogbrugen er en anden end i dag. Ex: Tidligere var ordet 'husbond' udbredt på landet i betydningen arbejdsgiver, nu bruges 'arbejdsgiver' både i byerne og på landet. Her kan en tesaurus hjælpe med at fastholde udviklingen i begrebernes betydning og dermed lette søgning i ældre

dokumenter.

Krav til et juridisk informationssøgesystem

Jurister står overfor et krav om ikke at overse en eneste relevant lov, lovændring eller lignende. En konsekvens af det er, at mange søgninger i Edb-Karnov eller et lignende system vil blive meget brede og omfatte meget store retsområder. I de fleste tilfælde giver den ekstra bredde kun irrelevante dokumenter; men der er ikke råd til at miste de få gange, hvor en helt uventet lov, paragraf, henvisning eller lignende dukker op. Et juridisk informationssøgesystem vil derfor blive mødt med tre hovedkrav: 1) Systemet skal være opdateret, 2) søgningerne skal være fuldstændige, og 3) det skal være let at få et overblik over, hvad systemet omfatter, og hvad der må søges andre steder. Til disse tre krav føjes et fjerde af mere generel karakter: 4) Systemet skal være meget let tilgængeligt - meget brugervenligt.

Et opdateret system. Jurister er helt afhængige af, at de lovtekster, de baserer sagsbehandlingen på, er gældende ret. Der går derfor megen tid med at kontrollere, om der eksisterer senere ændringer og tilføjelser i forhold til en given lovtekst. Det væsentligste krav til et juridisk søgesystem er ifølge Dorthe la Cour, at det overtager denne opgave. Kravet til et opdateret system er, at ændringer i lovteksten er lagt ind indenfor 24 timer. Hvis opdateringen ikke sker prompte, kan juristen ikke med sikkerhed basere sagsbehandlingen på systemet og skal derfor alligevel til at lave papirarbejde. Senere i interviewet foreslår Dorthe la Cour et alternativ til opdatering indenfor 24 timer: Opdatering et par gange om ugen kombineret med en tydelig angivelse af tidspunktet for sidste og næste opdatering. Det vil reducere papirarbejdet til perioden fra sidste opdatering frem til nu - hvis 'slippet' ikke er til at leve med, og man ikke kan vente på næste opdatering.

Fuldstændige søgninger. Jurister søger ofte svar på spørgsmål, der falder indenfor et af de fasttømrede begreber, som fx 'arbejdsskade' eller 'opsigelse'. I disse tilfælde vil en relevant tekst ofte indeholde dette ord. Det vil imidlertid ikke altid være tilfældet, og derudover er det naturligvis ikke alle sager, der kan karakteriseres ved et af de fasttømrede begreber. Mange søgninger vil således blive omfattende i et forsøg på at sikre, at alle relevante dokumenter findes frem. Problemet er ifølge Dorthe la Cour ikke, at de omfattende søgninger måske tager lang tid; men at der ikke er fuld sikkerhed for, at enhver relevant oplysning bliver fundet frem.

Problemet med fuldstændigheden opstår, fordi der er langt fra mening til formulering. Fuldtekstsøgning alene er utilstrækkeligt, og selvom en tesaurus hjælper, løser den ikke problemet. Tesaurusser arbejder typisk med enkeltord og kan derfor ikke afgøre, at fx 'underslæb' og 'ulovlig omgang med betroede midler' er synonymer. Juristernes krav om fuldstændige søgninger kan ikke honoreres; det er ikke muligt at få et *recall* på 100% uden at få en uacceptabelt lav *precision.* Det skyldes, at gevinster i *recall* opnåes på bekostning af *precision* (se figur 2.6). Af denne grund har Per Sjøqvist meget lidt tilovers for fuldtekstsøgning:

> *For folk, som har indsigt i det [et retsområde] i forvejen, er fuldtekstsøgning en forbandelse, en vederstyggelighed.*

Han begrunder denne afvisning af fuldtekstsøgning med, at man enten mister flere relevante dokumenter eller drukner i irrelevante. Dvs enten er *recall* for lavt eller også er *precision* det. Per Sjøqvist har i stedet stor tillid til nøgleord, fx i form af sagregistret i Karnovs Lovsamling. Men som det fremgår af kapitel 2 anses det generelt for endnu sværere at få et meget højt *recall* ved hjælp af nøgleordsbaseret søgning. Det fremgår

endvidere af kapitel 2, at flere mener, at gode søgesystemer må kombinere fuldtekstsøgning og faciliteter til nøgleordsbaseret søgning.

Systemets omfang. Hvis juristen er overbevist om, at svaret på et spørgsmål ikke findes i systemet, skal søgningen fortsættes andre steder. Det stiller krav om, at det er let at få et overblik over, hvilke kilder og tekster systemet har søgt i.

Tilgængelighed og brugervenlighed. Jurister er ikke indstillet på at ofre selve søgesystemet - redskabet - ret megen opmærksomhed. Det stiller meget store krav til enkelhed i menneske/maskine-samspillet. Kravet, om at søgesystemet er meget let tilgængeligt og brugervenligt, skyldes først og fremmest, at informationssøgning kun er en lille del af juristers arbejde. Jurister bruger langt størstedelen af deres tid på at etablere et personligt forhold til deres klienter, på juridiske forhandlinger og på andre møder (Leith 1990). Dorthe la Cour erkender, at der er en konflikt mellem de mange forskellige ting, juristerne gerne vil have systemet kan, og den opmærksomhed, de er villige til at ofre det.

3.2 Eksisterende hjælpemidler

De to mest omtalte juridiske søgesystemer er de to amerikanske, LEXIS og WESTLAW. De er begge privatejede foretagender og ligger i konstant konkurrence. LEXIS har været et fuldtekstsøgesystem siden starten i 1973. WESTLAW var nøgleordsbaseret indtil 1978; men da brugerne foretrak fuldtekstsøgning - de brugte LEXIS i stedet for WESTLAW - blev fuldtekstsøgning tilføjet (Sprowl 1981).

I Skandinavien har det offentlige spillet en central rolle i forbindelse med etableringen af juridiske søgesystemer. Før vi vender os mod Danmark, vil vi ud fra Bing (1984) give en meget kort oversigt over de skandinaviske søgesystemer til jurister. Sverige var et af foregangslandene i Europa. Allerede i 1966 undersøgte det svenske justitsministerium mulighederne for at etablere et juridisk informationssøgesystem. Det førte i første omgang til udviklingen af et system, der fungerede internt i forvaltningen. I 1980 blev systemet tilgængeligt for offentligheden under navnet Rättsdata. I Norge begyndte overvejelser om juridisk informationssøgning i 1970 på Institut for Retsinformatik. Overvejelserne resulterede i det juridiske fuldtekstsøgesystem Lovdata, der blev tilgængeligt i 1983. Lovdata er udviklet af et privat firma i samarbejde med det juridiske fakultet ved Oslo Universitet. I Danmark startede man med at udvikle et system på et afgrænset område. Datacentralen udviklede DC-jura, der dækkede det skatteretslige område. Derudover udgav Schultz Forlag DATA LEX, der ikke er et fuldtekstsøgesystem, men først og fremmest et register. I 1986 blev både DC-jura og DATA LEX erstattet af Retsinformation.

I dette afsnit vil vi beskrive to danske lovsamlinger. Først vil vi beskrive Karnovs Lovsamling. Karnovs Lovsamling er på bogform, men relevant her af to årsager: Den er en af de mest benyttede lovsamlinger i Danmark, og (en del af) den udgør datagrundlaget for den prototype, vores case skal munde ud i. Derefter vil vi beskrive Retsinformation. Retsinformation er et fuldtekstsøgesystem og i øjeblikket det eneste edb-baserede juridiske informationssøgesystem i Danmark.

Karnovs Lovsamling

Karnovs Lovsamling udgives af Karnovs Forlag og udkommer i et oplag på 10.000. Den er en institution i juridiske kredse. Det, der adskiller Karnovs Lovsamling fra andre lovsamlinger, er, at der er knyttet noter til lovteksterne. Karnovs Lovsamling er således den eneste generelle kommenterede lovsamling i Danmark (Blume 1989). Noterne skrives af en stab på omkring 180 højtkvalificerede jurister og består for en stor del af koncentrerede referater af lovforarbejderne og principielle domsafgørelser. Den følgende

beskrivelse af Karnovs Lovsamling er baseret på en artikel af professor dr. jur. W. E. von Eyben (1989), den ene af de

Figur 3.2. Eksempel på en side fra Karnovs Lovsamling (her med begyndelsen af lbkg 1977 nr 559 om mægling i arbejdsstridigheder). Den stiplede linie adskiller lovtekst og noter.

tre redaktører på Karnovs Forlag.

Karnovs Lovsamling består af tre dele: Lovtekster, noter og registre. *Lovteksterne* omfatter næsten alle gældende love (enkelte lokale, tidsbegrænsede og meget specielle love er udeladt), og desuden en del bekendtgørelser og de vigtigste cirkulærer. Lovteksterne medtages uændret i deres fulde ordlyd; alle tilføjelser og kommentarer sker i form af noter. Princippet for udarbejdelsen af *noterne* er, at alt, hvad der kan antages at have betydning, medtages. Selvom noterne er koncentrerede, udgør de ca halvdelen af teksten i Karnovs Lovsamling. I noterne refereres de forarbejder, fx betænkninger, der er gået forud for lovtekstens endelige udformning. Noterne indeholder også referater af principielle domsafgørelser og af det administrative stof, der bidrager til fortolkning og uddybning af lovteksten. Derudover indeholder noterne store mængder referencer til andre lovtekster. Figur 3.2 viser et eksempel på en side fra Karnovs Lovsamling.

Man finder rundt i Karnovs Lovsamling ved hjælp af fire *registre*: Emneregistret, det kronologiske register, det alfabetiske register og sagregistret. *Emneregistret* er en detaljeret indholdsfortegnelse, ordnet i overstemmelse med systematikken i Karnovs Lovsamling. Emneregistret er hierarkisk opbygget i fra tre til fem niveauer og velegnet til at give et overblik over hvilke lovtekster, der behandler et givet retsområde. Ex: Tjenestemandsloven kan findes gennem fire valg af ikke-overlappende retsområder:

1. Stats- og forvaltningsret
  E. Offentlig administration
    1. Almindelige regler om tjenestemandsforhold
      a. Løn- og ansættelsesvilkår
        Tjenestemandslov, **lbkg 1986** 671 ændret ved **L 1987** 885

Lovteksterne er altså ordnet systematisk i et hierarki. Det er ikke uden problemer, hvad redaktørerne på Karnovs Forlag da heller ikke lægger skjul på (von Eyben 1989):

> *Adskillige love hører efter deres natur hjemme i forskellige afsnit, hvorfor redaktionen må træffe et valg og iøvrigt gå ud fra, at man ved hjælp af registrene trods alt er i stand til at finde en lov, hvor den så end måtte blive placeret.*

I *det kronologiske register* er lovteksterne ordnet efter dato og nummer. I *det alfabetiske register* kan lovteksterne slås op ud fra de væsentligste ord i lovtekstens navn. Det fjerde og sidste register er *sagregistret*, der omfatter ca 15.000 stikord. Sagregistret er ordnet alfabetisk, men betjener sig i meget høj grad af hovedord. Det betyder fx, at et spørgsmål om særlig indkomstskat skal søges som et underpunkt under hovedordet 'skat'. Mens de tre foregående registre henviser til lovtekster, henviser sagregistret ofte direkte til paragraffer og eventuelt til noter. Det er således gennem sagregistret, man får de mest præcise henvisninger.

"Hvermands Lovbog" - forløberen for Karnovs Lovsamling - udkom første gang i 1924. Den indeholdt ingen noter; de kom til i 1938 med den tredje udgave af lovsamlingen. Gældende ret har ændret sig meget siden da og ændrer sig fortsat. En meget væsentlig del af arbejdet med Karnovs Lovsamling ligger derfor i ajourføringen.

Hvert tredje år udkommer en ny udgave af hovedsamlingen. Den består fra og med 11. udgave (1986) af tre bind. Den 11. udgave er på ialt 3707 tætskrevne sider, den 12. udgave (1989) på 4058 sider. I de to år ind imellem udgivelsen af to hovedsamlinger udkommer et årligt suppleringsbind med alle de nye eller ændrede lovtekster og dertilhørende noter. De to suppleringsbind til 11. udgave omfatter tilsammen 3351 sider, altså nogenlunde det samme som hovedsamlingens tre bind. Udover suppleringsbindene

udkommer såkaldte "grønne hæfter", normalt på 16 sider, hvor love, bekendtgørelser og cirkulærer løbende føres ajour. Ajourføringen af lovteksterne sker altså gennem suppleringsbindene og de "grønne hæfter". Det er imidlertid ikke blot lovteksterne, der ændrer sig, fortolkningen af dem ændrer sig også - typisk i form af ændringer i retspraksis eller administrativ praksis. Der er således ofte behov for ajourføringer i noterne, selvom den tilhørende lovtekst ikke er ændret. Ajourføringen af noterne til lovtekster, der ikke er ændret, sker i notetillægget.

For at det skal være muligt at finde den gældende lov indenfor et eller andet område og samtlige tilhørende noter, udgives hvert år et ajourført registerbind. Registerbindet indeholder de fire allerede omtalte registre, det ovenfor nævnte notetillæg samt en fortegnelse over ændrede og ophævede love. Et opslag i lovsamlingen bør således foregå ved først at finde lovteksten i et af registerbindets registre, derefter slå lovteksten op i det angivne hoved- eller suppleringsbind og endelig kontrollere, om notetillægget indeholder nye/ændrede noter. Man får kun det fulde udbytte af Karnovs Lovsamling, hvis man har sat sig ind i den noget komplicerede måde, ajourføringen foregår på. Det har imidlertid vist sig (von Eyben 1989), at selv en del mangeårige abonnenter på lovsamlingen ikke forstår dens opbygning fuldt ud og derfor går glip af oplysninger, de kunne have fået.

Karnovs Lovsamling er ikke et edb-system og opfylder da også kun ét af de fire krav, jurister stiller til edb-baserede juridiske informationssøgesystemer: Det er meget let at få et overblik over lovsamlingens omfang, fx gennem emneregistret. Til gengæld er Karnovs Lovsamling ikke opdateret; den måde, ajourføringen foregår på, reducerer tilgængeligheden betydeligt; og fuldstændige søgninger kan kun opnås ved manuel skimming.

Retsinformation

Retsinformation er statens juridiske edb-informationssøgesystem. Grundlaget for systemet er to betænkninger fra Retsinformationsrådet, betænkning nr 1001/1984 om en lovdatabase og betænkning nr 1144/1988 om databaser med konkrete afgørelser. Retsinformation planlægges og koordineres af Sekretariatet for Retsinformation under Justitsministeriet. Størstedelen af den tekniske ajourføring varetages af Schultz Forlag. Redaktionen og ajourføringen af systemets dokumenter påhviler imidlertid de enkelte informationsleverandører - først og fremmest de 22 ministerier. Endelig er systemet placeret på I/S Datacentralen, som også varetager brugeradministrationen. Retsinformation blev åbnet for søgning i februar 1986 og har ca 3.000 tilsluttede brugere (januar 1991). Det følgende er dels baseret på (Blume 1989) og (Retsinformation 1989), dels på en demonstration af systemet.

Retsinformation er et fuldtekstsøgesystem og adskiller sig primært fra de andre kilder til information om de juridiske forhold i Danmark ved sit omfang og de løbende ajourføringer. De løbende ajourføringer betyder, at nye/ændrede lovtekster er tilgængelige i Retsinformation senest samtidig med deres offentliggørelse i Lovtidende eller lignende publikationer.

Retsinformation indeholder en imponerende mængde data. Dokumenterne i systemet er inddelt i tre grupper: Lovforarbejder, lovtekster og andet. *Lovforarbejderne* dækker det arbejde, der går forud for vedtagelsen af lovene, og bruges ofte til - på et senere tidspunkt - at prøve at afgøre, hvad der var intentionen med loven. Lovforarbejderne omfatter samtlige taler i Folketinget og samtlige lovforslag og betænkninger siden folketingsåret 1982/83. Det er fx muligt at finde alt, hvad undervisningsministeren sagde fra Folketingets talerstol i foråret 1989 i forbindelse med SU-reformen.

Den anden gruppe - *lovteksterne* - omfatter love, bekendtgørelser og cirkulærer. Næsten alle gældende lovtekster er indlagt i systemet; mængden af bekendtgørelser og

specielt cirkulærer er således meget større end i Karnovs Lovsamling. Når en lovtekst erstattes af en anden, slettes den ikke fra systemet; den overføres derimod til systemets afdeling for forældede regler. Retsinformation kan således også bruges i forbindelse med ældre sager, der skal afgøres ud fra retsgrundlaget på det pågældende tidspunkt.

Den tredje gruppe - *andet* - omfatter domme, kendelser og administrative afgørelser. Denne del af Retsinformation er vedtaget, men endnu ikke etableret. Det er intentionen, at alle principielle domme, kendelser og afgørelser skal lægges ind i systemet. På nuværende tidspunkt indeholder Retsinformation kun en række skatteretslige afgørelser, som alle stammer fra DC-jura.

Dokumenterne i Retsinformation er organiseret i baser. Vi nøjes med at se på organiseringen af den mest omfattende gruppe - lovteksterne. Opdelingen af lovteksterne i baser omfatter dels en opdeling efter, hvor lovteksten stammer fra (en opdeling efter informations-leverandør), dels en opdeling efter lovtekstens type. Lovteksterne er opdelt i en base for hvert ministerium, og hver af disse baser er underopdelt i tre baser: En lovbase, en bekendtgørelsesbase og en cirkulærebase. Det er muligt at afgrænse søgningerne til ét ministeriums baser eller søge i alle ministeriers baser på en gang. Uafhængigt af det kan søgningerne afgrænses til en enkelt af de pågældende ministeriers baser, til deres lov- og bekendtgørelsesbaser på en gang eller til alle deres tre baser på en gang. En søgning kan derimod ikke afgrænses til landbrugs- og miljøministeriet og heller ikke til bekendtgørelser og cirkulærer.

Søgning i Retsinformation forløber i tre niveauer. Først vælges den eller de baser, søgningen skal omfatte. Søgningen kan fx afgrænses til Justitsministeriets lovbase. Derefter kan brugeren udvælge en række lovtekster ved at angive en forespørgsel. Forespørgslen kunne fx specificere alle lovtekster, hvor ordene 'statslig' og 'retsinformation' (eller ord der kan afskæres til 'statslig' henholdsvis 'retsinformation') forekommer i samme sætning. På det sidste niveau vælger brugeren hvilke af de fremfundne dokumenter, der skal vises i fuld tekst. Indenfor de viste dokumenter kan brugeren igen benytte søgning til at finde den paragraf eller det stykke, der er relevant. Brugeren kan fx vælge at få alle de fremfundne dokumenter vist og søge efter forekomster af ordet 'regelsanering'. Eksemplerne ovenfor vil give anledning til følgende søgekommandoer:

```
BASE JLOV
FIND statslig* s retsinformation*
VIS
SØG regelsanering
```

I afsnit 3.1 pegede vi på fire krav, som jurister stiller til søgesystemer. Retsinformation lever op til kravet om at være opdateret. Da systemet indeholder så at sige enhver lovtekst, er det endvidere let at få et overblik over, hvilke dokumenter, Retsinformation omfatter, og hvilke der må søges andre steder. Derimod er søgningerne ikke fuldstændige, da fuldtekstsøgningen ikke støttes af en tesaurus eller lignende. Endelig er Retsinformation ikke særlig brugervenligt. Ex: Lovteksterne er organiseret efter informationsleverandør i stedet for efter behovene i brugssituationen, og det er ikke muligt at specificere, hvilke ministerier en søgning skal omfatte - der er kun mulighederne ét eller alle. Der er yderligere to væsentlige minusser ved systemet: Det er for dyrt at bruge, og det er forbudt at kopiere data fra Retsinformation over i fx et tekstbehandlingssystem. Efter vores mening giver edb væsentligt bedre muligheder for at støtte brugeren, end det kommer til udtryk i Retsinformation.

3.3 Udvikling over tid

Hvis et søgesystem skal fungere tilfredsstillende, skal det efter vores mening være orienteret mod brugssituationen. Vi fokuserer i dette projekt på et fundamentalt træk ved brugssituationen: Den ændrer sig løbende. Vi ser et stort behov for faciliteter, der giver mulighed for, at disse ændringer indarbejdes i søgesystemet, så snart brugeren bliver opmærksom på dem. Behovet for søgesystemer, der er konstrueret med henblik på udvikling over tid, udspringer af, at det på et tidspunkt i udviklingen af systemerne er nødvendigt at fastfryse kravene til dem. Tidspunktet for fastfrysningen af kravene er meget forskelligt indenfor de traditionelle systemarbejdsmetoder - fasemodellerne - og eksperimentel systemudvikling; men den finder sted i begge tilfælde (Hertzum & Søes 1990). Brugssituationen ændrer sig imidlertid også efter, at kravene er fastfrosset. Vi mener derfor ikke, det er tilstrækkeligt at inddrage udvikling over tid i systemudviklingen. Det er nødvendigt at inddrage og søge at indbygge det i selve systemet.

I det omfang det lykkes at indbygge udvikling over tid i systemet, vil det lette vedligeholdelsen af systemet. Det er lettere for en systemudvikler at ændre i et program, hvis det er udviklet med henblik på ændringer. Det primære formål er imidlertid at give brugeren mulighed for i tilknytning til sit arbejde med systemet at foretage de ændringer og justeringer, der udgør systemets udvikling over tid. Store ændringer, fx at udvide et fuldtekstsøgesystem med faciliteter til nøgleordsbaseret søgning, skal naturligvis foretages af systemudviklere. Det er imidlertid de mindre ændringer - dem der foretages fra dag til dag - vi fokuserer på. Vi kan skelne mellem to situationer, hvor et systems muligheder for at udvikle sig udnyttes:

- En statisk situation, hvor systemet skal kunne tilpasses til forskellige brugere med hver deres individuelle behov og præferencer.
- En dynamisk situation, hvor systemet skal kunne udvikle sig over tid for vedblivende at tilfredsstille den enkelte bruger.

Den statiske situation er på ingen måde irrelevant; men det er den dynamiske, vi har i tankerne i vores diskussion af udvikling over tid. Omverdenen, organisationen og sprogbrugen ændrer sig, og dermed ændres kravene til søgesystemet også. Det betyder, at ændringer i fx sprogbrug skal kunne indarbejdes i søgesystemet, såsnart brugeren bliver opmærksom på dem. Ellers vil brugeren gang på gang opleve systemet som utidssvarende og forstyrrende for arbejdet med de opgaver, søgningerne skulle bidrage til at løse.

Som et eksempel på hvordan sprogbrugen ændrer sig, har vi tidligere nævnt, at ordet 'arbejdsgiver' nu har erstattet ordet 'husbond'. I andre tilfælde opstår nye begreber som betegnelser for nyskabelser i verden omkring os. I 1989-udgaven af registerbindet til Karnovs Lovsamling findes der således et begreb, som ikke findes i de tidligere udgaver. Der blevet tilføjet en ny indgang under hovedordet 'ægteskab' i sagregistret; denne indgang indeholder en henvisning til det nye hovedord 'registreret partnerskab'. De to ovenfor nævnte eksempler på ændringer/nyskabelser i sprogbrugen kan indføres i et søgesystem i form af nye relaterede begreber i tesaurussen. I nogle tilfælde vil disse tilføjelser være foretaget i den næste udgave af systemet; men udbyderen har ingen mulighed for at opspore og forudse alle de begreber, brugerne anvender og gerne vil have lagt ind i tesaurussen. Vi mener derfor, det er afgørende, at brugeren selv har mulighed for at foretage sådanne ændringer i tesaurussen.

Grupperingen af søgesystemets dokumenter er et andet eksempel på en facilitet, hvor brugeren med fordel kan have mulighed for at foretage løbende ændringer. I Retsinformation er lovteksterne organiseret efter informationsleverandør, dvs ressortministerium. Det er uhensigtsmæssigt, dels fordi brugerne ikke tænker i ressortministerier, men i retsområder, dels fordi dele af lovgivningen fra tid til anden

skifter ressortministerium. Denne uhensigtsmæssighed forstærkes af, at det ikke er muligt at specificere, hvilke ministerier en søgning skal omfatte - der er kun mulighederne ét eller alle. I Karnovs Lovsamling udgør emne-registret en hierarkisk opdeling af lovteksterne i retsområder. Redaktørerne lægger imidlertid ikke skjul på, at opdelingen giver nogle problemer, og at placeringen af nogle lovtekster er lidt tilfældig (se afsnit 3.2). Vi ser et behov for et dynamisk emneregister organiseret i retsområder. Med 'dynamisk' mener vi, at den enkelte bruger kan indsætte, rette og slette i emneregistret. Et dynamisk emneregister giver brugeren mulighed for at gruppere lovteksterne efter sine egne kriterier og øjeblikkelige behov. Specielt kan brugeren samle alle de tekster, der er relevante for en given type sager, i ét særskilt retsområde. Alle søgninger i forbindelse med denne type sager kan derefter foregå indenfor dette retsområde.

3.4 Fuldtekstsøgning overfor nøgleordsbaseret søgning

Både fuldtekstsøgning og nøgleordsbaseret søgning har, som det fremgår af afsnit 2.3, deres styrker og svagheder. Sprowl (1981) betragter det da også som en fordel, at WESTLAW omfatter såvel fuldtekstsøgning som faciliteter til nøgleordsbaseret søgning. Af de to eksempler, vi har behandlet i afsnit 3.2, benytter Retsinformation udelukkende fuldtekstsøgning, mens sagregistret i Karnovs Lovsamling svarer til nøgleord. De forskellige styrker og svagheder ved fuldtekstsøgning og nøgleordsbaseret søgning er imidlertid ikke tilfældigt fordelt; vi mener at kunne pege på en fundamental forskel mellem de situationer, de er velegnede i:

Nøgleordsbaseret søgning bygger på en antagelse om, at det indekserede område altid er velstruktureret. Strukturen kan være implicit eller eksplicit; men tilstedeværelsen af en struktur er en forudsætning for, at der kan bestemmes en præcis og dækkende mængde nøgleord for hver tekst. Hvis hver tekst behandler få og velafgrænsede emner, og hvert emne kun behandles i få tekster, så er nøgleordsbaseret søgning velegnet: Der er gode muligheder for at opnå såvel højt *recall* som høj *precision*. Hvis teksterne ikke opfylder dette krav, opstår der problemer. Et af disse problemer er, at fastlæggelsen af en teksts nøgleord får det til at se ud, som om den kun omhandler en begrænset og veldefineret mængde emner.

Fuldtekstsøgning forudsætter ikke nogen struktur og forsøger heller ikke at definere nogen. Ved fuldtekstsøgning lægges der ingen systematik ind mellem brugeren og teksterne, til gengæld er der heller ingen systematik at basere søgningerne på. Fuldtekstsøgning er således velegnet, når søgningerne ikke følger teksternes hovedlinier, men går på tværs eller skal lokalisere detaljer. En tværgående søgning kunne fx være en søgning efter metodeovervejelser i en samling af videnskabelige artikler. Derudover er fuldtekstsøgning selvfølgelig velegnet i forbindelse med tekstsamlinger, hvor der ikke med rimelighed kan defineres en systematik.

Lovgivningssystemet er usædvanligt velstruktureret. Strukturen, som udgør et vidtforgrenet hierarki, er grundlaget for den juridiske metode (se afsnit 3.1). Det ligger derfor lige for at støtte den juridiske metode ved at inkludere faciliteter til nøgleordsbaseret søgning i et juridisk informationssøgesystem. Den juridiske metode kombineret med nøgleordsbaseret søgning udgør en *top-down* søgestrategi. Per Sjøqvist foretrækker denne søgestrategi og mener, den dækker en jurists behov. Vi kan godt se fordelene ved denne måde at søge på; men vi mener ikke, den er tilstrækkelig.

Fuldtekstsøgning er ikke en dårlig realisering af ideen i nøgleordsbaseret søgning; det er en anden måde at søge på. For at tydeliggøre forskellen vil vi - selvom betegnelsen måske er lidt upræcis - sige, at fuldtekstsøgning giver mulighed for en *bottom-up* søgestrategi. Med fuldtekstsøgning er det muligt at søge uafhængigt af strukturen i

lovgivningssystemet og uden at passe forespørgslen ind efter de etablerede juridiske begreber, der er optaget som nøgleord. Ex: En sådan søgning kunne være en søgning efter lovtekster, der vedrører et givet geografisk område, fx Grønland. Det er utænkeligt, at områdets navn optræder som nøgleord i forbindelse med alle de lovtekster, der gælder for området. Selve lovteksten må imidlertid formodes at indeholde oplysninger om, hvorvidt den gælder eller netop ikke gælder for det pågældende område.

Vi mener, at der er behov for at kombinere fuldtekstsøgning og nøgleordsbaseret søgning. Et godt søgesystem bør omfatte begge muligheder. I dette projekt ligger hovedvægten på fuldtekstsøgning. Som støtte til fuldtekstsøgningen indeholder Edb-Karnov en tesaurus. Vi ser en oplagt mulighed for også at bruge begreberne i denne tesaurus som nøgleord. I forbindelse med Edb-Karnovs tesaurus vil vi derfor give mulighed for en enkel form for nøgleordsbaseret søgning.

3.5 Introduktion til Edb-Karnov

Emnet for vores case er design og konstruktion af Edb-karnov - en edb-udgave af Karnovs Lovsamling. Edb-Karnov er en prototype på et juridisk informationssøgesystem med fokus på faciliteter, der tillader systemet at udvikle sig over tid. Vi afviser tanken om 'ekspert-søgesystemer'. I stedet udvikles Edb-Karnov med henblik på at fungere som et redskab for fagfolk. Datagrundlaget for prototypen er godt 400 sider fra Virksomheds-Karnov (ca 4 Mb tekst). Virksomheds-Karnov er et udtræk af Karnovs Lovsamling, hvor de lovtekster, der er relevante for erhvervslivet, er samlet. Lovteksterne og de tilhørende noter fremstår i Virksomheds-Karnov fuldstændig, som de gør i Karnovs Lovsamling. Registrene er ligeledes de samme, men omfatter selvfølgelig kun de indgange, der er relevante for de inkluderede lovtekster.

Juristers krav til juridiske informationssøgesystemer kan sammenfattes i fire hovedpunkter (se afsnit 3.1). Det første er kravet om, at systemet skal være opdateret. Dette krav er rettet mod ajourføringen og dermed indirekte mod systemets redigeringsfunktioner. I prototypen vil vi ikke beskæftige os med ajourføringen af lovteksterne; Edb-Karnov er således ikke et opdateret system. I kapitel 5 vender vi imidlertid tilbage til ajourføringen, idet vi vil undersøge, hvordan ajourføringen - udgivelsen af nye udgaver af Edb-Karnov - kan integreres med de ændringer, brugeren foretager løbende. Det er primært de tre andre krav, vi kommer ind på i vores case. De tre andre krav foreskriver, at søgningerne skal være fuldstændige, at det skal være let at få et overblik over systemets omfang, og at systemet skal være meget let tilgængeligt. Disse krav er rettet mod søgningerne og de funktioner, den enkelte jurist har til rådighed i sin brug af systemet.

Edb-Karnov er en prototype og omfatter derfor nogle, men ikke alle, de faciliteter, et juridisk informationssøgesystem efter vores mening bør omfatte. Blandt de afgrænsninger, vi har foretaget, er en af de væsentlige, at Edb-Karnov er udviklet som et enkeltbrugersystem. Et fuldt udbygget system skal fungere som et flerbrugersystem i et netværk. Vi vil gennem hele udviklingen af Edb-Karnov diskutere vores prototype op imod *state of the art* for fuldtekstsøgesystemer og det, et fuldt udbygget juridisk søgesystem efter vores mening bør kunne.

Edb-Karnov er et fuldtekstsøgesystem, hvor søgningerne foregår ved hjælp af boolsk søgning med nærhedsoperatorer. For at gøre implementeringen overkommelig, indskrænker vi os til én nærhedsoperator. Brugeren kan således angive, at to ord skal forekomme 'indenfor samme paragraf'. Efter vores mening er det ikke tilstrækkeligt, at et søgesystem tilbyder enten fuldtekstsøgning eller nøgleordsbaseret søgning (se afsnit 3.4). Edb-Karnov indeholder derfor også en facilitet til en enkel form for nøgleordsbaseret

søgning.

Vi vil behandle tre faciliteter, der har til formål at støtte Edb-Karnovs udvikling over tid. Den første er et dynamisk emneregister. Emneregistret giver mulighed for at opdele databasen i klynger. En klynge kan være et retsområde eller en hvilken som helst anden opdeling, brugeren måtte ønske. Det er endvidere muligt at afgrænse søgningerne til en given klynge. Den anden er en dynamisk tesaurus. Brugeren har mulighed for at slette, rette og tilføje ord, således at tesaurussen kan yde den bedst mulige støtte i valget af søgeord. Det er endvidere i tesaurussen, faciliteten til nøgleordsbaseret søgning er placeret; ordene i tesaurussen kan således også bruges som nøgleord. Den tredje facilitet, der har til formål at støtte udvikling over tid, er egne notater. Egne notater giver brugeren mulighed for at føje nye tekster til systemet, og de er søgbare på fuldstændig samme måde som Edb-Karnovs øvrige tekster.

3.6 Sammenfatning

I dette kapitel har vi søgt at give et indtryk af den brugssituation, vores case udspiller sig i. Kapitlet er mundet ud i en første beskrivelse af Edb-Karnov. Vi har beskrevet jurister som en gruppe af fagfolk. Juristen, sagen og hjælpemidlerne i form af lovsamlinger og lignende indgår i et komplekst samspil. Det er dette samspil, vi benævner juridisk sagsbehandling. Et nyt hjælpemiddel, fx et fuldtekstsøgesystem, vil ændre dette samspil og dermed påvirke den måde, juristen oplever sagen på. Det er således svært at vurdere et nyt hjælpemiddel, før det har været i brug et stykke tid. Et nyt hjælpemiddels værdi kommer ikke fuldt ud til udtryk, sålænge det vurderes ud fra det eksisterende samspil mellem juristen, sagen og de etablerede hjælpemidler. Dette forbehold gælder også vores indkredsning af juristers krav og forventninger til juridiske informationssøgesystemer.

Lovgivningssystemet er ret velstruktureret, og de enkelte lovtekster er i vid udstrækning formuleret som en række omstændigheder med tilhørende konsekvenser. Det betyder imidlertid ikke, at den mening, lovteksterne indeholder, kan formaliseres og repræsenteres i et edb-system. For at forstå lovteksterne kræves fortolkning. Denne fortolkning må inddrage såvel den sammenhæng, de enkelte paragraffer indgår i, som den sammenhæng, lovteksterne tænkes anvendt i. Juridisk sagsbehandling kan således ikke automatiseres. På grund af de enorme mængder tekst, der muligvis indeholder relevante informationer, vil et godt søgesystem imidlertid kunne yde juristen værdifuld støtte.

Vi har lokaliseret fire krav, som et juridisk informationssøgesystem vil blive mødt med: 1) Systemet skal være opdateret, 2) søgningerne skal være fuldstændige, 3) det skal være let at få et overblik over systemets omfang, og 4) systemet skal være meget let tilgængeligt. Det fremgår, at jurister stiller store krav til juridiske informationssøgesystemer. Specielt kravet om fuldstændige søgninger kan ikke honoreres fuldt ud. Hvadenten et søgesystem tilbyder fuldtekstsøgning, nøgleordsbaseret søgning eller begge dele, vil brugeren ikke kunne opnå et *recall* på 100% uden at få en uacceptabelt lav *precision*.

Efter i dette kapitel at have behandlet de rammer, Edb-Karnov skal fungere i, vil vi i næste kapitel vende os mod konstruktionen og implementeringen af vores prototype på et juridisk fuldtekstsøgesystem. Hensigten med prototypen er at udvikle og evaluere vores ideer om faciliteter, der tillader søgesystemer at udvikle sig over tid. Prototypen skal således være idéudviklende og kravspecificerende. Systemets primære søgeteknik bliver boolsk søgning med en enkelt nærhedsoperator. Et centralt problem ved fuldtekstsøgning er, at søgeordene skal forekomme i de relevante dokumenter, hvis disse dokumenter skal findes frem.For at reducere dette problem støttes søgningerne af såvel et emneregister som en tesaurus. Brugeren kan ændre i emneregistret og tesaurussen efter behov og har

endvidere mulighed for at føje egne notater til de tekster, systemet allerede indeholder.

4. Design og konstruktion af Edb-Karnov

Dette kapitel handler om Edb-Karnov - den case vi har valgt til at illustrere og afprøve vores ideer om fuldtekstsøgesystemer, der er indrettet på at udvikle sig over tid. I slutningen af forrige kapitel gav vi en kort introduktion af Edb-Karnovs funktioner og faciliteter; i dette kapitel diskuteres og fastlægges hver enkelt. Vi implementerer ikke et fuldstændigt system, men en tilstrækkelig stor del til at muliggøre en realistisk vurdering af vore ideers holdbarhed. Kapitlet munder således ud i en prototype, hvor der er gjort meget ud af nogle funktioner, mens andre slet ikke er med. Årsagen til dette er naturligvis behovet for at begrænse og målrette projektet.

Formålet med afgrænsningerne er dels at reducere projektets omfang dels at gøre behandlingen af Edb-Karnovs centrale dele klar og af en vis generel relevans. Af den grund vil vi i løbet af kapitlet afgrænse os fra en del af de problemstillinger, der er specifikke for udviklingen af en edb-udgave af Karnovs Lovsamling. I det følgende sammenfattes de centale dele i Edb-Karnov - de dele af et fuldstændigt system vi vil lægge vægt på:

- *Generelt*. Edb-Karnov skal primært muliggøre fuldtekstsøgning; men det skal også omfatte begrænsede muligheder for nøgleordsbaseret søgning. Endvidere er Edb-Karnov et søgesystem, ikke et redigeringssystem.
- *Søgning*. Edb-Karnov skal give mulighed for boolsk søgning ved hjælp af OG, ELLER og nærhedsoperatorer samt give mulighed for opslag ud fra nøgleord. Derudover skal det være muligt at bladre frem og tilbage i de fundne tekster, at følge notehenvisninger og at gå fra en note til det sted, hvor der henvises til den.
- *Præsentation og import/eksport*. De fundne tekster skal præsenteres i vinduer, og det skal være muligt frit at bevæge sig fra vindue til vindue. Derudover skal det være muligt at eksportere tekst (en paragraf, en note eller et notat) til fx et tekstbehandlingssystem og importere tekst fra et tekstbehandlingssystem til et personligt notat.
- *Udvikling over tid*. Edb-Karnov skal indeholde faciliteter, der tillader systemet at udvikle sig over tid. Det skal være muligt at ændre i den systematik, der giver en oversigt over Edb-Karnovs dokumenter; at tilpasse søgesystemet til ændringer i sprogbrug og begrebsdefinitioner; og at tilføje egne notater til søgesystemet.

Kapitlet er opbygget således, at der sker en trinvis udvikling af Edb-Karnov - først de grundliggende faciliteter, så emneregistret, derefter tesaurussen og endelig egne notater. Det betyder blandt andet, at datamodellen for vores prototype udvikles og udvides hen igennem kapitlet. Den endelige datamodel er gengivet sidst i kapitlet (se figur 4.21). Udviklingen af denne datamodel forløber i følgende afsnit:

1. Relationsdatabaser som grundlag for søgesystemer
2. Valg af værktøjer
3. Grundlaget for fuldtekstsøgning
4. Boolsk søgning og de grundliggende skærmbilleder
5. Dynamisk emneregister
6. Dynamisk tesaurus
7. Egne notater
8. Afprøvning og optimering
9. Sammenfatning

4.1 Relationsdatabaser som grundlag for søgesystemer

Det fremgår af afsnit 2.2, at brugen af relationsdatabaser som grundlag for informationssøgesystemer har været genstand for megen kritik. Essensen i kritikken er, at søgesystemer baseret på relationsdatabaser er for langsomme og fylder for meget. Vi vil alligevel basere konstruktionen af Edb-Karnov på en relationsdatabase. Det er der flere grunde til:

For det første mener vi, pladsforbruget er underordnet i forbindelse med et system som Edb-Karnov. Udviklingen indenfor lagringsteknologierne har gennem en årrække betydet, at stadigt større lagre blev tilgængelige til stadigt lavere priser. I løbet af de seneste år er lagre med meget stor kapacitet blevet ganske billige. Ex: I 1980 kostede en 5 Mb harddisk ca 45.000 kr excl moms. I dag (maj 1991) sælges en 300 Mb henholdsvis 1,2 Gb harddisk til en Macintosh til 22.800 henholdsvis 51.300 kr excl moms. I slutningen af 80'erne blev de optiske diske også en mulighed; udviklingen indenfor dette område beskrives i (Burke & Ryan 1989). De ikke-sletbare optiske diske - CD-ROM og WORM ('Write Once Read Many times') - benyttes allerede i en del søgesystemer, og nu begynder de sletbare optiske diske at vise sig på markedet. Vi kan og vil ikke afvise kritikken af, at søgesystemer baseret på relationsdatabaser kræver store mængder lagerplads. Vi mener imidlertid ikke, pladskravene er for store. Søgesystemer baseret på relationsdatabaser fylder meget; men det er efter vores mening underordnet i forbindelse med Edb-Karnov og lignende systemer, da prisen på lagerplads efterhånden er lav og fortsat kraftigt faldende.

For det andet mener vi, effektive relationsdatabasesystemer nu er blevet tilgængelige. Det betyder ikke, at problemet med svartiderne er elimineret; men det er reduceret. Det er endvidere vigtigt at vurdere svartiderne i forhold til de krav, brugerne sætter til dem. Små og enkle søgninger skal naturligvis give anledning til korte svartider, ellers opfattes søgningen som hæmmende for brugerens arbejde. For de omfattende og komplekse søgninger er vi imidlertid overbevist om, at brugernes vurdering af svartiden afhænger meget af deres oplevelse af hvor stor en søgning, de har sat i gang; af værdien af resulatet; og af den tid, det ville tage at udføre en tilsvarende søgning manuelt. Derfor er svartiden ikke nær så kritisk i disse tilfælde; der er andre forhold, som spiller en større rolle. Dorthe la Cour siger direkte, at det er fuldstændigheden, ikke svartiden, der er kritisk. Hun er således villig til at acceptere tunge søgninger, hvis hun til gengæld får en meget høj sikkerhed for ikke at gå glip af et relevant dokument. Der er endvidere en lang række muligheder for at optimere relationsdatabaser, fx kan der ofte opnåes markante fald i svartiderne ved at gå fra tredje til anden normalform. På denne baggrund er vi ikke overbevist om, at søgesystemer baseret på relationsdatabaser er for langsomme. På grund af deres funktionalitet og fleksibilitet fortjener brugen af relationsdatabaser i hvert fald at blive efterprøvet.

For det tredje og afgørende mener vi, som det fremfår af afsnit 2.2, at relationsdatabaser har en funktionalitet og fleksibilitet, der giver Edb-Karnov gode muligheder for at leve op til kravet om udvikling over tid. Den enkle model og datauafhængighederne gør systemet let at ændre og udvide. Specielt giver muligheden for at oprette *views* gode muligheder for at ændre i databasens tabeller, uden at det er nødvendigt at ændre i applikationerne. Relationsdatabasernes funktionalitet og fleksibilitet skyldes, at de er udviklet til en bred vifte af formål. Prisen for denne funktionalitet og fleksibilitet er større pladskrav og længere svartider, end det er tilfældet med de søgesystemer, der er udviklet specielt med henblik på tekstsøgning. Dette skal imidlertid ses i sammenhæng med, at relationsdatabaser, så vidt vi ved, er det eneste grundlag for søgesystemer, der kan støtte udvikling over tid.

Ud fra anbefalingerne i litteraturen ville det være naturligt at basere Edb-Karnov på inverterede filer. Vi kan sammenfatte vores argumentation for alligevel at basere Edb-

Karnov på en relationsdatabase i tre punkter: For det første kan man ikke gøre noget med inverterede filer, som man ikke også kan gøre med relationsdatabaser; men det koster typisk mere tid og plads at bruge relationsdatabaser. For det andet får man nogle nye muligheder med relationsdatabaser, primært en væsentligt større fleksibilitet. For det tredje er fleksibilitet afgørende i forbindelse med vores krav om, at Edb-Karnov skal kunne udvikle sig over tid. Vi vurderer derfor, at relationsdatabasernes fordele mere end opvejer deres ulemper.

Relationsdatabaser er ikke udviklet til én bestemt type anvendelser. Herved adskiller de sig markant fra de søgesystemer, de typisk sammenlignes med. Relationsdatabasernes generalitet har sin pris, men giver også anledning til at tro, at der kan opnåes meget ved et omhyggeligt systemdesign.

4.2 Valg af værktøjer

Edb-Karnov er et PC-baseret system. Det giver først og fremmest to muligheder i valget af datamat: En traditionel PC'er eller en Macintosh. Vi skal bruge et relationsdatabasesystem, et generelt programmeringssprog og gerne et værktøj til udvikling af brugergrænseflader. Kravene til værktøjerne er, at de er effektive og veldokumenterede, og at der er fornuftige grænseflader mellem dem.

Til datamater i PC-størrelse findes der ifølge Finkelstein & Pascal (1988) seks kommercielt tilgængelige relationsdatabasesystemer: Informix, Ingres, Oracle, SQLBase, XDB II og XQL. Finkelstein & Pascal's tests giver ikke anledning til, at de fremhæver nogle fremfor andre; de konkluderer blot, at alle seks systemer har deres styrker og svagheder. Vi har valgt at udvikle Edb-Karnov i Oracle på en Macintosh. Det skyldes, at DIKU råder over Oracle til Macintosh'erne, og at vi ønskede at prøve at arbejde med Macintosh'en og dens avancerede grafiske brugergrænseflade. Vi er senere blevet bekendt med Matos & Jalics (1989). De sammenligner svartiderne for forskellige relationsdatabasesystemer til PC'ere og finder, at Oracle er meget langsommere end de øvrige databasesystemer, specielt ved *joins* af store tabeller. Vi kendte ikke denne vurdering af Oracle, da vi valgte databasesystem. Hvis den viser sig også at dække Oracle på Macintosh, får vi grundlag for at sige, at de resultater, vi når med Oracle, ligeledes kan opnåes med andre relationsdatabasesystemer.

Vi har desuden behov for et generelt programmeringssprog, fx til at læse tekst fra Virksomheds-Karnov ind i databasen. Oracle har en *precompiler*, der gør det muligt at indlejre SQL-kommandoer i C-programmer. Valget af Oracle betyder derfor, at C er det mest nærliggende valg af generelt programmeringssprog. Oracle understøtter endvidere kommunikation med Hypercard, et af Macintosh'ernes værktøjer til udvikling af brugergrænseflader. Det er nemt at udvikle skærmbilleder i Hypercard, så de følger standarden for Macintosh'ens grafiske brugergrænseflade. Vi vælger altså Macintosh (model IIcx), Oracle (version 1.2.0), C (MPW C version 3.0) og Hypercard (version 1.2.5) til implementeringen af Edb-Karnov.

4.3 Grundlaget for fuldtekstsøgning

I dette afsnit diskuteres og fastlægges grundstammen i Edb-Karnov. Afsnittet drejer sig om, hvordan lov- og noteteksterne skal lagres, hvordan søgning skal gøres mulig, og hvordan notehenvisningerne skal håndteres. Den del af datamodellen for Edb-Karnov, som fastlægges i dette afsnit, er vist i figur 4.1, sidst i afsnittet.

Lagring af tekst

Da Edb-Karnov er et fuldtekstsøgesystem, skal selve lov- og noteteksten lægges ind i systemets tabeller. Lovteksterne omfatter love, lovbekendtgørelser, bekendtgørelser og cirkulærer. Love og lovbekendtgørelser er strukturerede og ens opbygget, mens der er en vis variation i opbygningen af bekendtgørelser og cirkulærer. Forskellene mellem de forskellige typer lovtekster spiller imidlertid ingen rolle for lagringen - de kan uden problemer lagres i samme tabel. Noterne adskiller sig derimod væsentligt fra lovteksterne. Den væsentligste forskel er, at lovteksterne er selvstændige enheder, mens noterne altid er knyttet til (et bestemt sted i) en lovtekst. Vi adskiller derfor lovteksterne og noterne i to tabeller. Fordelen ved denne adskillelse er først og fremmest, at notehenvisningerne bliver lettere at håndtere.

Før lovteksterne kan lægges ind i Edb-Karnov, skal der vælges en grundenhed for dem. En almindelig tekst kan typisk opdeles hierarkisk i kapitler, afsnit, tekstafsnit og sætninger. Lovtekster har en lidt anden, men meget gennemført og eksplicit struktur med hele teksten, paragraffen og stykket som de tre mest fasttømrede niveauer. Vi mener, fire valg af grundenhed kan komme på tale: Linier, stykker, paragraffer og lovtekster.

Linier. Vælges linier, vælges implicit også fast linielængde. Opdeling i linier er imidlertid meget knyttet til tekstens udseende på bogform; i et edb-system opløses bogformens tætte afhængighed mellem lagringen og præsentationen af teksten. I et edb-system, hvor teksten præsenteres i vinduer af varierende størrelse, har linier således ingen mening i forbindelse med lagringen af teksten. Vi vil derfor ikke vælge linier som grundenhed for lagringen.

Stykker. Argumentet for at vælge stykker er, at stykket er lovtekstens mindste logiske enhed. Stykket er den mindste tekstdel, der har et selvstændigt juridisk indhold, og det er den mindste enhed, der henvises til - fra andre lovtekster og fra juridisk materiale iøvrigt. Et stykkes betydning er imidlertid meget afhængigt af den kontekst, stykket forekommer i. Dorthe la Cour udtrykker afhængigheden mellem stykket og konteksten - først og fremmest paragraffen - meget tydeligt:

> *En af de første ting, man lærer på universitetet [på jurastudiet] er, at man aldrig må læse stykke 3 uden at have læst stykke 1 og 2. Af den grund, at stykke 1 er hovedreglen og så kommer undtagelserne. Og hvis du ikke har fattet hovedreglen, så forstår du heller ikke undtagelsen, der står i stykke 3.*

På denne baggrund mener vi, at stykker er utilstrækkelige i præsentationssammenhæng. Det er nærliggende at vælge den samme grundenhed i lagringen og præsentationen; men det er selvfølgelig ikke nødvendigt. Hvis andre forhold taler for det, kan teksten lagres som stykker og præsenteres flere stykker ad gangen. Vi kan se én fordel ved at vælge stykker som grundenhed for lagringen: Alle henvisninger kan repræsenteres med fuld præcision. Da teksten vil blive præsenteret i en større enhed end stykker, er der efter vores mening kun tale om en meget lille fordel. Denne fordel skal ses i sammenhæng med, at det er mere tidskrævende at hente flere lagrede enheder frem, hver gang noget skal præsenteres, end blot at hente én. Vi vil se bort fra de fuldstændige henvisninger til fordel for at få den samme grundenhed i lagringen og præsentationen. Dette valg hænger nøje sammen med, at Edb-Karnov er et søgesystem; i et redigeringssystem vil man sandsynligvis prioritere omvendt.

Paragraffer. Det fremgår implicit af citatet ovenfor, at paragraffer er et bedre valg af grundenhed for præsentationen. Jens Peter Nielsen har overfor os givet udtryk for, at de to vigtigste niveauer i trykte såvel som edb-baserede lovsamlinger er love og paragraffer. Paragrafferne er lovteksternes grundenhed, og i hvert fald ét af en lovsamlings registre skal henvise til paragraffer. Han mener, erfarne jurister - i vidt omfang - kan nøjes med at

blive henvist til lovtekster, da de selv hurtigt kan finde de relevante paragraffer. Alle andre er afhængige af at blive henvist direkte til de relevante paragraffer. Dette er efter vores mening gode argumenter for at vælge paragraffer som grundenhed for lagringen. Vi finder det endvidere rimeligt at repræsentere henvisninger til stykker ved henvisninger til den paragraf, stykket indgår i.

Lovtekster. Vi vælger ikke lovtekster som grundenhed. Valgtes lovtekster, ville det lægge op til, at en lovtekst præsenteres ad én gang, og at søgning indenfor den enkelte lovtekst i vid udstrækning foregår ved bladring. Lovteksterne er efter vores mening alt for omfattende til, at dette er hensigtsmæssigt. Per Sjøqvist giver imidlertid udtryk for, at han foretrækker denne løsning pånær i de situationer, hvor der søges i de meget store lovtekster. Det skyldes, at han ikke har tillid til, at fuldtekstsøgning er finmasket nok til at arbejde med mindre enheder end lovtekster.

Vi har valgt paragraffer som grundenhed for lagringen af teksten. Det næste spørgsmål er, hvordan en paragraf identificeres. Indenfor en lov identificeres paragrafferne entydigt ved et nummer eventuelt kombineret med et litra (et bogstav der bruges som betegnelse for fx en paragraf), fx §13 og §11a. Brugen af litraer skyldes for en stor del, at jurister husker placeringen af store mængder lovstof ved lovtekstens navn og paragrafnummeret; tilføjelse af en ny paragraf må derfor ikke føre til, at alle efterfølgende paragraffer omnummereres. Paragrafferne i en lov kan være samlet i afsnit og/eller kapitler. Denne opdeling kan man muligvis have glæde af at repræsentere eksplicit i datamodellen; vi vil imidlertid undlade at behandle den yderligere. En paragraf identificeres således entydigt i hele Edb-Karnov ved dens nummer, dens litra og den lovtekst, den indgår i.

Oracle giver mulighed for at lagre tekststrenge på indtil 64 Kb i en særlig type variable. Der er visse begrænsninger på, hvilke operationer der kan udføres med disse variable; først og fremmest kan de ikke indgå i *where*-delen af SQL-sætningerne. Vi finder dem alligevel velegnede til lagring af paragraffernes tekst. Det skyldes, at al søgning foregår ved hjælp af andre tabeller, først og fremmest søgeordstabellen. Den eneste operation, der skal udføres på disse særlige variable, er fremfinding med henblik på præsentation. Lovteksterne kan således lagres i en tabel med fire felter: lovteksten paragraffen indgår i, paragraffens nummer, dens litra og selve paragrafteksten.

Vores valg af, hvordan lovteksterne skal lagres, har kun givet anledning til ét problem. Dette problem skyldes, at vi meget nødig vil ændre i selve lovteksten. Det har vi imidlertid været nødt til at gøre i et par tilfælde, der alle svarer til det følgende:

> I momsloven (lbkg 1988 nr 629) står:
>
> *41-44. - - - (Udeladt).*
>
> For at gøre alle paragraffer søgbare, har vi erstattet ovenstående med:
>
> *41. - - - (Udeladt).*
> *42. - - - (Udeladt).*
> *43. - - - (Udeladt).*
> *44. - - - (Udeladt).*

Søgeord

Da Edb-Karnov er et fuldtekstsøgesystem, skal alle meningsbærende ord i teksten være søgbare. Det opnåes ved at basere søgningerne på en søgeordstabel, genereret ved invertering af den lagrede tekst. Søgeordstabeller er et kompromis; det, brugerne er interesserede i, er de relevante dokumenter, ikke de dokumenter hvor bestemte ord forekommer. En mulighed for at forbedre en søgeordstabel er at udvide den med en række

faste vendinger. Vi finder denne mulighed udfordrende, men meget tidskrævende, og afgrænser os derfor fra den. Hver tupel i søgeordstabellen skal indeholde et ord og en entydig henvisning til det sted, hvor ordet forekommer, dvs en tupel består af et ord, en lovidentifikation, et paragraf-nummer og et paragraflitra.

I afsnit 3.5 afgrænsede vi os til én nærhedsoperator: 'Indenfor samme paragraf'. Hvis Edb-Karnov yderligere skulle omfatte nærhedsoperatoren 'indenfor samme stykke', skulle søgeordstabellens henvisninger til ordenes forekomster udvides med en attribut, der angiver stykket. Vi gør opmærksom på, at dette valg kan træffes uafhængigt af valget af grundenhed for lagringen af teksten. Det skyldes, at det under inverteringen af teksten er en simpel sag at holde styr på hvilket stykke, ordene forekommer i.

Før søgeordstabellen kan dannes, skal det fastlægges, hvad der udgør et ord. Ashford (1987) foreslår, at ord defineres ved deres grænser. Han benytter en mængde af starttegn og en mængde af sluttegn; et ord defineres derefter som det, der står mellem et starttegn og det følgende sluttegn. Vi har defineret ord på en anden måde, nemlig ved de tegn, de indeholder. Ord består af bogstaver og eventuelt en bindestreg, der hverken må være ordets første eller sidste bogstav. Ved at definere ord ved deres indhold fremfor deres grænser opnår vi en strammere kontrol med, hvad der genkendes som ord; det har vi vurderet som en fordel.

En speciel type ord opstår, når to ord skrives sammen til ét ved hjælp af en bindestreg, fx 'edb-register' og 'Virksomheds-Karnov'. Disse ord optages i søgeordstabellen. For brugeren af et søgesystem vil det imidlertid ofte være et gæt, hvorvidt et ord er stavet med bindestreg eller ud i ét. Ex: Det hedder 'Virksomheds-Karnov' med bindestreg, selvom bindes normalt betyder, at ordet skrives uden bindestreg. For at reducere dette problem optages de to ord, som disse sammenskrevne ord består af, også i søgeordstabellen. Ex: Ordet 'edb-register' giver anledning til søgeordene 'edb-register', 'edb' og 'register'. Det er således både muligt at lave en snæver søgning på det sammenskrevne ord og at søge bredere på de enkelte delord. Den største fordel ved dette er, at en søgning med det ene søgeord 'edb' vil finde en tekst med ordet 'edb-register', selvom ordet 'edb' ikke forekommer i den pågældende tekst.

For at lette implementeringen lidt afgrænser vi os til at tage højde for ord, der indeholder én bindestreg. Der er imidlertid enkelte ord, der opstår ved at sammenskrive mere end to ord ved hjælp af bindestreger; vi har fx fundet '4-ugers-perioden' og 'dag-til-dag' i lovteksterne. Disse ord optages ikke i søgeordstabellen. De delord, som disse sammenskrevne ord består af, optages imidlertid i søgeordstabellen. Ex: '4-ugers-perioden' giver anledning til søgeordene 'ugers' og 'perioden'; '4' er et tal og optages derfor ikke i søgeordstabellen.

Som regel er det en fordel, hvis søgningerne er ufølsomme overfor variationer i brugen af store og små bogstaver. I enkelte tilfælde afhænger et ords betydning imidlertid af brugen af store og små bogstaver; vi har tidligere brugt eksemplet 'ROM', 'Rom' og 'rom'. Vi mener, der er ret få af disse tilfælde, og tillader os derfor at afgrænse os fra at behandle dem yderligere. Alle ord konverteres således til udelukkende små bogstaver, før de lægges ind i søgeordstabellen.

Vores definition af ord og de øvrige valg i det ovenstående er præget af Edb-Karnovs anvendelsesområde. Juridiske tekster består så at sige udelukkende af ord skrevet med alfabetets 28 bogstaver, af tal og af de gængse tegn til tegnsætningen (punktum, komma osv). Paragraftegnet er en væsentlig undtagelse; men juridiske tekster indeholder meget få specialtegn og andre grafiske elementer. Hvis anvendelsesområdet havde været et andet, skulle definitionen af, hvad der er et ord, også have været en anden. Det er fx ikke nødvendigt at bevæge sig over i matematiske eller tekniske tekster for at finde ord, der består af såvel bogstaver som cifre. Ex: I Edb-Karnov kan såvel TV2 (TV-stationen) som

TV2 (rock-gruppen) kun søges ved hjælp af søgeordet 'tv'. Da der endvidere ikke skelnes mellem store og små bogstaver, vil denne søgning også finde alle de steder, hvor 'tv' bruges i betydningen 'til venstre'.

Stopliste

Søgeordstabellen skal ikke indeholde alle de forekomne ord, men kun de af dem, en bruger kan finde på at anvende som søgeord. Da der er en lille gruppe ikke-meningsbærende ord, som forekommer meget hyppigt, kan en stopliste reducere antallet af tupler i søgeordstabellen væsentligt. Edb-Karnovs stopliste omfatter 324 ord og er gengivet i bilag 2.

Stoplisten er udviklet i to trin. Først fik vi via Bo Nielsen, Dansk Arbejdsgiverforening, den danske standardstopliste til BRS - et standardsystem til udvikling af fuldtekstsøgesystemer. Denne liste omfatter 120 stopord, men er underlagt den begrænsning, at den ikke må indeholde mere end 128 ord. Vores stopliste er en tabel i en relationsdatabase og derfor ikke underlagt en lignende begrænsning. Det andet trin bestod i at udvide stoplisten med ord fra Virksomheds-Karnov. Vi valgte 50 sider i Virksomheds-Karnov og udskrev alle de forekomne ord sammen med den frekvens, de forekommer med. Derefter gennemgik vi denne liste for ord, som med sikkerhed ikke var meningsbærende og forekom med en vis frekvens. Til slut udvidede vi stoplisten med alle ord på ét bogstav, 'ø' og 'å' undtaget. Det resulterede i 204 nye stopord.

En stopliste bruges udelukkende for at reducere antallet af tupler i søgeordstabellen, herved spares lagerplads, og det tager mindre tid at afgøre, om et ord forekommer i en tekst. Normalt betyder en stopliste, at antallet af tupler i søgeordstabellen reduceres med 40-50% (Salton & McGill 1983). I Edb-Karnov har reduktionen været på 39%. Det tyder på, at der ikke kan opnåes ret meget ved at udvide stoplisten yderligere: Der kan formodentlig føjes mange ord til Edb-Karnovs stopliste; men de forekommer ikke så ofte, at det vil reducere størrelsen af søgeordstabellen væsentligt.

Notehenvisninger

Ovenfor har vi primært diskuteret ud fra lovteksten. Virksomheds-Karnov indeholder imidlertid også noter, og det er i meget vid udstrækning dem, der er baggrunden for udbredelsen af Karnovs udgivelser. Noterne er knyttet til lovteksterne ved henvisninger, der kan se ud på tre forskellige måder: '*)', 'nr)' eller '(nr)'. Når disse tegnfølger optræder i lovteksterne, er der altid tale om notehenvisninger. Notehenvisningerne i lovteksterne er blot én type henvisninger; der er også henvisninger mellem lovteksterne samt fra noterne til såvel lovteksterne som andre noter. Der ligger en mulighed i at gøre enhver henvisning i teksterne til en funktion, som ved aktivering fører til, at der slåes op på den tekst, der henvises til. Mulighederne for at indlægge sådanne hypertekstfaciliteter i en edb-udgave af Virksomheds-Karnov er undersøgt af Jensen (1990). Vi vil ikke inddrage hypertekstfaciliteter og afgrænser os til at behandle lovteksternes notehenvisninger.

Det skal naturligvis være muligt at slå en note op, når der henvises til den i lovteksten. Den væsentligste grund til, at vi behandler notehenvisningerne, er imidlertid, at de spiller en vigtig rolle i søgningerne: Brugeren kan fx ønske at søge i noterne til momsloven eller at se alle de paragraffer, hvor enten paragraffen selv eller en tilhørende note indeholder ordet 'byggeri'. Notehenvisningerne skal altså repræsenteres i datamodellen på en sådan måde, at det både er muligt at gå fra lovteksten til noterne og den modsatte vej.

Når der i en lovtekst henvises til en note, sker det ved angivelse af notens nummer og litra, hvis den har et litra. Ved at lagre disse to oplysninger gøres det muligt at gå fra note-henvisning til note, indenfor en lov. For at gøre det muligt at gå den anden vej - fra note til notehenvisning - skal notehenvisningens placering i lovteksten også lagres. Noterne er

ofte knyttet til stykker i lovteksten, i nogle tilfælde endda til enkelte ord; som tidligere diskuteret vil vi blot betragte en note som knyttet til en af lovtekstens grundenheder. Notehenvisningernes placering lagres altså i form af en oplysning om, hvilken paragraf henvisningen forekommer i.

I langt de fleste tilfælde er der én henvisning til hver note; men enkelte steder i Virksomheds-Karnov er der flere henvisninger til samme note (der er også enkelte noter, der ikke henvises til). Da vi finder dette punkt perifert i forhold til projektets emne, tillader vi os en afgrænsning: Vi vil ikke tillade, at der i flere forskellige paragraffer forekommer henvisninger til samme note. I de tilfælde, hvor sådanne henvisninger forekommer, behandles den første normalt, og der ses bort fra de øvrige. I det følgende eksempel foreslås en simpel måde at bevare alle henvisningerne på. Ex: Den anden og tredje henvisning til note 14 erstattes af henvisninger til henholdsvis note 14a og note 14b begge med ordlyden 'se note 14'.

Datamodel

Vi afslutter dette afsnit med en kort beskrivelse af datamodellen, som den ser ud på dette tidspunkt. De seks tabeller, som dette afsnit resulterer i (se figur 4.1), udgør grundstammen i Edb-Karnov. I de følgende afsnit vil vi føje nye tabeller til denne grundstamme.

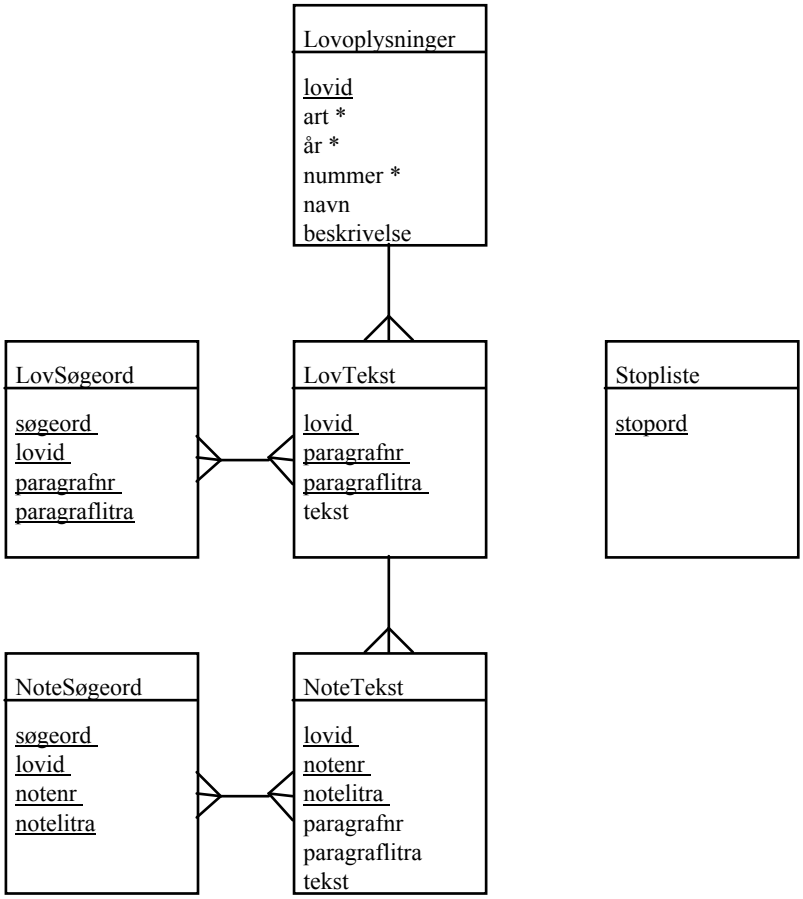
Lovteksterne giver anledning til tre tabeller. *Lovtekst* og *lovsøgeord* er diskuteret ovenfor; *lovoplysninger* bruges til at knytte nogle identifikationsoplysninger til hver lovtekst. Disse oplysninger skal primært bruges i præsentationen af de lovtekster, en søgning er resulteret i; men de giver også mulighed for opslag på lovteksterne svarende til det kronologiske register i Virksomheds-Karnov. Oplysningerne omfatter lovtekstens art (lov, lovbekendtgørelse, bekendtgørelse eller cirkulære), år, nummer, navn og beskrivelse. Som navne benytter vi korte, mundrette navne, typisk taget fra Virksomheds-Karnovs spalteoverskrifter, fremfor lovteksternes officielle juridiske navne. Beskrivelsesattributen er en lidt længere angivelse af, hvad lovteksten handler om. Selvom fuldtekstsøgesystemer giver mulighed for, at de fundne tekster læses fra skærmen, bruges de ofte kun til søgningen. Når de relevante tekster er fundet, findes en papirudgave frem til læsningen. For at lette denne proces kunne det overvejes at udvide *lovoplysninger* med det sidenummer i Virksomheds-Karnov, hvor lovteksten begynder.

Noterne giver anledning til to tabeller: *Notetekst*, hvor grundenheden er en note, og *notesøgeord*. Disse to tabeller er opbygget på samme måde som de tilsvarende tabeller for lovteksterne, pånær at der til notetekstabellen er tilføjet, hvilken paragraf noten er knyttet til. Endelig er *stopliste* blot en tabel af tupler med én attribut.

I modellen bliver tekst og søgeord lagret i to sæt tabeller, et for henholdsvis lovtekster og noter. Vi vælger først og fremmest denne opdeling for at undgå meget store tabeller, der forøger svartiderne betragteligt. En tabel for lovtekster og en anden for noter betyder desuden, at oplysningen om, hvor notehenvisningen forekommer, kan lagres sammen med noten. Hvis lovtekster og noter blev slået sammen i én tabel, ville det for at opretholde tredje normaform være nødvendig med en særskilt tabel, der til hver note knyttede den paragraf, hvor notehenvisningen forekom.

For at reducere risikoen, for at databasen bliver inkonsistent i forbindelse med en opdatering, er det vigtigt, at den overholder normalformerne. Dette skal imidlertid vejes op imod, at overholdelse af normalformerne fører til flere tabeller, og dermed til flere *joins* og længere svartider. Vi vil her som i de følgende afsnit bruge normalformerne som rettesnore, ikke som ufravigelige krav. Vores arbejde med og bestemmelse af normalformerne er baseret på de enkle begreber og formuleringer i (Kent 1983). De seks tabeller, der udgør grundstammen i Edb-Karnovs datamodel, er alle på såvel tredje som

fjerde normalform.



Figur 4.1. Grundstammen i Edb-Karnovs datamodel. Internt i Edb-Karnov identificeres hver lovtekst ved et entydigt nummer - et lovid. Derudover benyttes sammensatte nøgler. Tabellernes primærnøgle er markeret ved understregning, alternative nøgler med stjerner.

4.4 Boolsk søgning og de grundliggende skærmbilleder

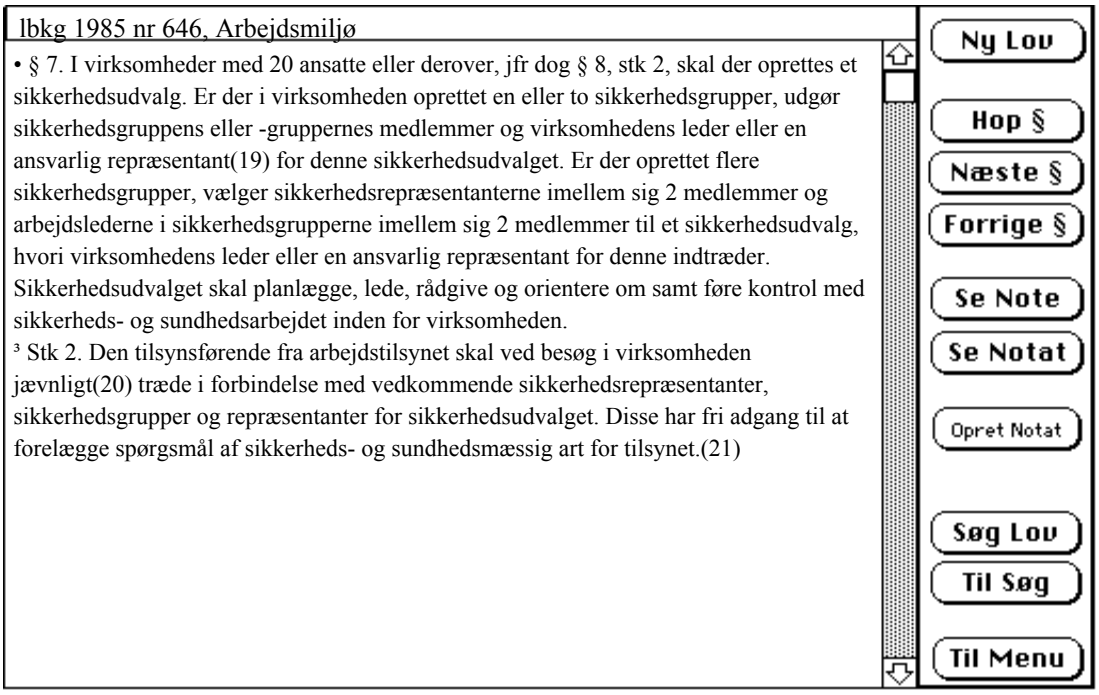
Edb-Karnov er en Hypercard-applikation ovenpå en Oracle-database. Mens det forrige afsnit var rettet mod grundstammen i selve databasen, er dette afsnit rettet mod de grundliggende dele af Hypercard-applikationen. I dette afsnit behandles de faciliteter, der implementeres ud fra grundstammen i datamodellen. Det drejer sig om de grundliggende søgefaciliteter og de fire grundliggende skærmbilleder - ét til Edb-Karnovs hovedmenu, ét til at vise lovteksterne, ét til at vise noterne og ét til at angive forespørgslerne og vise resultatet af søgningerne. Edb-Karnov har en grafisk brugergrænseflade; men hensigten med vores prototype er ikke at analysere og forfine brugergrænsefladen - det afgrænser vi os fra. Det, vi vil med Edb-Karnov, er at udvikle og vurdere vores ideer til faciliteter, der støtter udvikling over tid. Flere af de faciliteter, der behandles i dette afsnit, blev allerede berørt i forrige afsnit, da grundstammen i datamodellen blev fastlagt.

Afsnittet indledes med en kort beskrivelse af de fire grundliggende skærmbilleder og sammenhængen mellem dem. Her søger vi - i direkte forlængelse af forrige afsnit - at give et indtryk af, hvordan Edb-Karnov ser ud. Derefter diskuteres og beskrives Edb-Karnovs grundliggende søgefaciliteter. Det omfatter først valget af søgeteknik, derefter behandles forespørgslerne og til sidst præsentationen af de fremfundne tekster.

De fire grundliggende skærmbilleder

Edb-Karnov består af en række skærmbilleder, der hver er et kort i en Hypercard-stak. Heri ligger to væsentlige begrænsninger: For det første giver Hypercard (version 1.2.5, den ene-ste Oracle kan kommunikere med) ikke mulighed for applikationer med flere kort åbne samtidig. For det andet er kortstørrelsen fast; størrelsen af kortene er fastlagt sådan, at et kort netop fylder den mindste skærm, der leveres til Macintosh'erne. På den Macintosh IIcx, vi bruger, ligger omkring halvdelen af skærmen således ubrugt hen. Disse begrænsninger betyder, at Edb-Karnovs grundliggende skærmbilleder er hele skærmbilleder, ikke blot vinduer. Det betyder specielt, at en lovtekst og de tilhørende noter ikke kan vises samtidig; der er ikke plads til dem begge på ét kort, og der kan ikke vises to kort på én gang. I Edb-Karnov må der skiftes skærmbillede, når en note slås op. Det er en alvorlig begrænsning, da brugeren må formodes ofte at være interesseret i de noter, der er knyttet til lovteksten.

Det første af de grundliggende skærmbilleder er menuskærmbilledet. Det indeholder en oversigt over Edb-Karnovs øvrige skærmbilleder og muligheden for at skiftetil et vilkårligt af dem. Det er endvidere muligt at skifte til menu-skærmbilledet fra alle de øvrige skærmbilleder, således at det er let at komme rundt mellem de forskellige skærmbilleder. De tre øvrige grundliggende skærmbilleder er lovtekst- og note-skærmbilledet, der bruges til præsentation af Edb-Karnovs tekster i deres fulde ordlyd, og søge-skærmbilledet. Søge-skærmbilledet bruges til brugerens angivelse af forespørgsler og til systemets præsentation af den indledende oversigt over de fremfundne dokumenter. Lovtekst-, note- og søge-skærmbilledet beskrives kort i det følgende, først lovtekst-skærmbilledet (se figur 4.2).

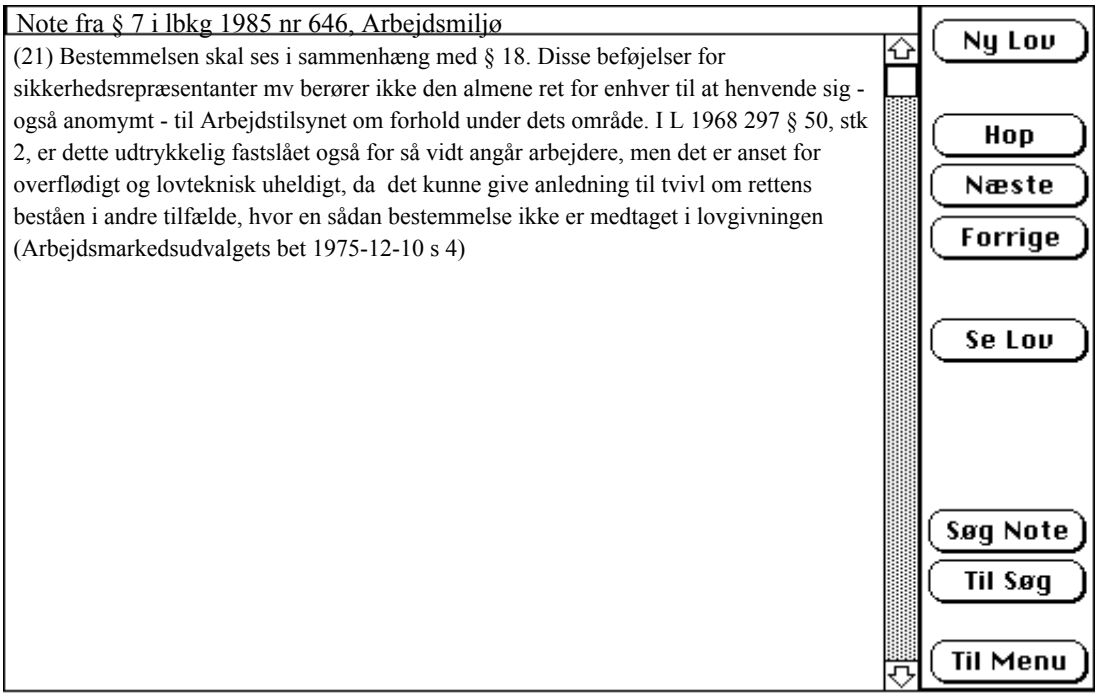


Figur 4.2. Lovtekst-skærmbilledet. På dette skærmbillede vises paragraffer fra de fremfundne lovtekster (her §7 i lbkg 1985 nr 646 om arbejdsmiljø). Med funktionerne til højre er det muligt at slå op på en ny lovtekst, at bladre i lovtekstens paragraffer, at slå en af paragraffens noter op og at skifte til søge- eller menu-skærmbilledet. Endelig kan der oprettes og slås op på notater; det vender vi tilbage til i afsnit 5.7.

Præsentationen af de fremfundne lovtekster sker i lovtekst-skærmbilledet, én paragraf ad

gangen. Da præsentationen sker én paragraf ad gangen, er der behov for funktioner til at skifte til en anden paragraf. Edb-Karnov giver såvel mulighed for at hoppe til en given paragraf som for at bladre frem til den næste eller tilbage til den forrige paragraf. Opslag på en ny lovtekst er egentlig en (meget simpel) søgefacilitet; men for at gøre denne funktion lettilgængelig findes den både på søge- og lovtekst-skærmbilledet. Opslaget kan ske enten ved at indtaste den nye lovteksts art, år og nummer eller ved at vælge den fra en kronologisk oversigt over alle systemets lovtekster. Fra lovtekst-skærmbilledet er det endvidere muligt at slå en af den viste paragrafs noter op. Det sker med funktionen Se Note, der viser en oversigt over de noter, der er knyttet til paragraffen. Fra denne oversigt vælges den ønskede note ved at klikke på den, herved skiftes til note-skærmbilledet. Endelig er det muligt at skifte til søge- eller menu-skærmbilledet. Et skift til søge-skærmbilledet kan foretages for at vende tilbage til oversigten over de tekster, den sidste søgning resulterede i; her bruges funktionen Til Søg. Skiftet kan også ske med henblik på at foretage en søgning indenfor den lovtekst, der i øjeblikket vises på lovtekst-skærmbilledet; i så fald er det lettest at bruge funktionen Søg Lov, der udover at skifte skærmbillede også initialiserer søgningen til kun at omfatte denne tekst.

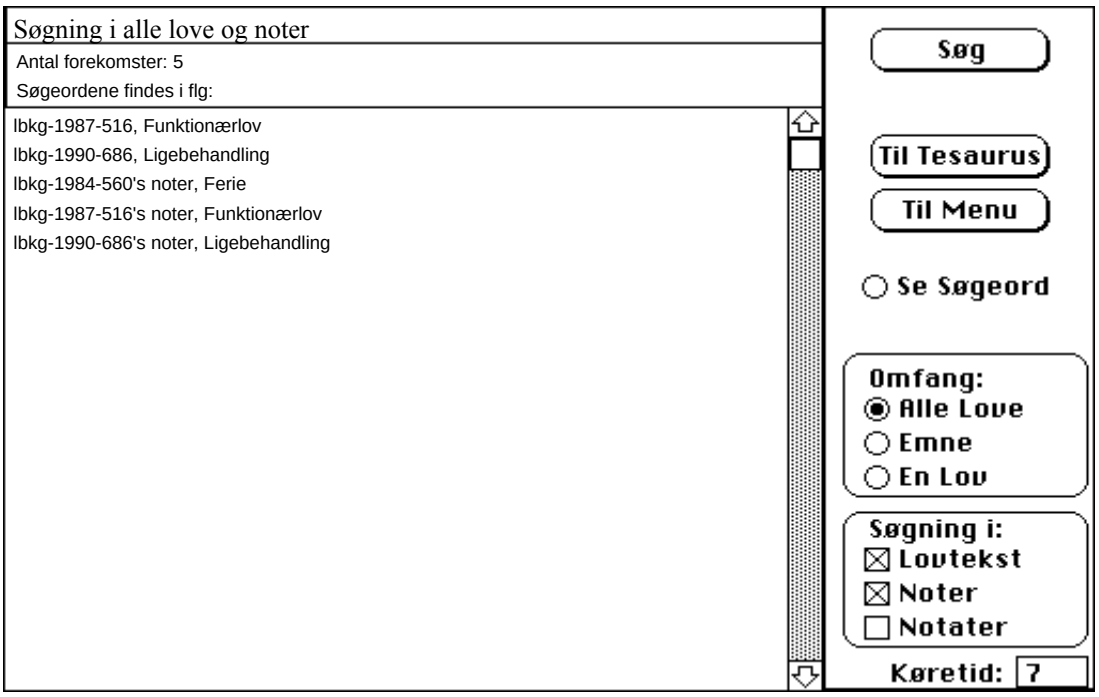
Note-skærmbilledet (se figur 4.3) svarer stort set til lovtekst-skærmbilledet. Den væsentligste forskel er, at funktionen Se Note her er erstattet af Se Lov. Se Lov skifter fra note-skærmbilledet til lovtekst-skærmbilledet, hvor den paragraf, der henviser til noten, bliver fundet og vist. I søgninger, der også omfatter noterne, gør denne funktion det umiddelbart muligt at få præsenteret de henvisende steder i lovteksterne.



Figur 4.3. Note-skærmbilledet (her med note 21 til §7 i lbkg 1985 nr 646 om arbejdsmiljø). Med funktionerne til højre er det muligt at skifte til en anden lovteksts noter, at bladre i noterne, at slå den henvisende paragraf op og at skifte til søge- eller menu-skærmbilledet.

Det fjerde grundliggende skærmbillede er søge-skærmbilledet (se figur 4.4). Det er på søge-skærmbilledet, brugeren angiver sine forespørgsler og søgningernes omfang. Det er også her, systemet viser oversigterne over hvilke tekster, der er fundet frem som svar på forespørgslerne. Fra oversigterne kan brugeren vælge en tekst og ved at klikke på den få den præsenteret i fuld tekst på enten lovtekst- eller note-skærmbilledet. Søgeteknikken

bag forespørgslerne og præsentationen af de fremfundne tekster behandles i det følgende.



Figur 4.4. Søge-skærmbilledet (her med resultatet af søgningen 'barsel% og fyring% eller afskedigelse%' i alle lovtekster og noter). Med funktionerne til højre er det muligt at udføre søgningen, skifte til tesaurus-skærmbilledet (se afsnit 4.6) eller menu-skærmbilledet, vise/ikke vise forespørgslen og fastlægge søgningens omfang. Muligheden for at afgrænse søgningen til et emne behandles i afsnit 4.5; notater behandles i afsnit 4.7.

Valg af søgeteknik

I det følgende behandles de grundliggende søgefaciliteter, der ligger bag søge-skærmbilledets funktioner; senere kommer to væsentligt mere avancerede til - emneregistret og tesaurussen. Vores første indkredsning af hvilke søgefaciliteter, Edb-Karnov skal stille til rådighed, tager udgangspunkt i de tre typiske sager, vi beskrev i afsnit 3.1. Det drejer sig om opslag, fremadrettet sagsbehandling og bagudrettet sagsbehandling. Med hensyn til opslag er der - som beskrevet ovenfor - mulighed for at slå en lovtekst op ved udpegning i en kronologisk oversigt eller ved angivelse af dens art, år og nummer. Derefter kan specifikke paragraffer slås op, ligesom der kan slås op på en af lovtekstens noter. Til disse faciliteter kan umiddelbart tilføjes en alfabetisk oversigt over systemets lovtekster svarende til det alfabetiske register i Virksomheds-Karnov; vi undlader imdlertid at implementere denne oversigt. Disse faciliteter giver direkte adgang til alle Edb-Karnovs lovtekster, paragraffer og noter og dækker efter vores mening behovene i forbindelse med opslag.

Med hensyn til fremad- og bagudrettet sagsbehandling er der imidlertid behov for egentlige søgefaciliteter. De forskellige muligheder for opslag er ikke tilstrækkelige. Disse sager må formodes at give anledning til en meget interaktiv søgeproces, hvor forespørgslerne ofte reformuleres på baggrund af en hurtig vurdering af de fremfundne tekster. Det betyder, at det skal være let at ændre lidt på en forespørgsel og derefter 'gentage' søgningen, og at søgningerne i første omgang skal resultere i en kortfattet oversigt over de fremfundne tekster. I afsnit 2.4 beskrev vi tre søgeteknikker, der kan komme på tale i forbindelse med et fuldtekstsøgesystem: Boolsk søgning, udvidet boolsk søgning og skimming.

Skimming er uegnet som den primære søgeteknik i et juridisk informationssøgesystem. Det skyldes, at det med skimming ikke er muligt at stille forespørgsler. Kombineret med en anden søgeteknik kan skimming imidlertid være værdifuldt. Den mest oplagte anvendelse af skimming i Edb-Karnov ville være i forbindelse med notehenvisningerne; men skimming kunne fx også bruges til at give direkte adgang til alle de lovtekster og noter, der henviser til den tekst, brugeren i øjeblikket har på skærmen. Som nævnt i forrige afsnit har Jensen (1990) undersøgt, om og hvordan skimming kan bruges i en edb-udgave af Virksomheds-Karnov; vi vil ikke komme yderligere ind på det.

En anden mulighed i valget af søgeteknik er udvidet boolsk søgning. Denne søgeteknik er efter vores mening ikke brugbar i praksis i sin nuværende udformning. Udvidet boolsk søgning er baseret på en antagelse om, at brugeren i vid udstrækning kan sætte tal på vigtigheden af forespørgslens forskellige dele. Denne kvantificering har form af vægte, som brugeren kan knytte til såvel søgeordene som de boolske operatorer. Formuleringen af en forespørgsel er imidlertid en proces, hvor en tilstand af utilstrækkelig viden skal formaliseres og repræsenteres. Der kan ikke med nogen rimelighed sættes tal på de søgeord og boolske operatorer, denne proces resulterer i. Vægtene skyldes udelukkende et behov for parametre, der er tilgængelige for beregning, først og fremmest med henblik på at rangordne de fremfundne dokumenter. Vi forkaster udvidet boolsk søgning og mister derved muligheden for at ordne de fremfundne dokumenter ud fra andet end dato og lignende.

Den søgeteknik, Edb-Karnov stiller til rådighed, er således boolsk søgning. Boolsk søgning kan udformes som forespørgsler formuleret med de boolske operatorer OG, ELLER og IKKE; det giver mulighed for at supplere forespørgslerne med nærhedsoperatorer. En anden mulighed er at udforme boolsk søgning som begrebsbaseret fremfinding (se afsnit 2.4). Her søges den formalisering og repræsentation, der fører frem til forespørgslen, lettet ved, at forespørgslen betragtes som en fællesmængde af flere begreber. Begrebsbaseret fremfinding er en mere brugervenlig overbygning på boolsk søgning, og det tror vi er god idé. Til gengæld begrænses variationsmulighederne til de forespørgsler, der kan formuleres udelukkende med OG og ELLER. Vi mener, at nærhedsoperatorer er relevante i forbindelse med Edb-Karnov. Vi kunne implementere både boolsk søgning i den almindelige udformning og begrebsbaseret fremfinding. Derved kunne brugeren fra gang til gang vælge den mulighed, der passede bedst. Vi afgrænser os fra begrebsbaseret fremfinding for at gøre implementeringen af søgefaciliteterne overkommelig; men vi betragter en udvidelse af Edb-Karnov med begrebsbaseret fremfinding som meget relevant.

Forespørgslen - angivelsen af søgningen

Søgningerne foretages fra søge-skærmbilledet ved angivelse af en forespørgsel og en række parametre, der bruges til at fastlægge søgningens omfang, fx om søgningen kun omfatter lovtekster eller også noter. Vi begynder med forespørgslerne. Forespørgslerne består af søgeord og operatorer. Søgeordene vælger brugerne frit, mens der kun er et begrænset antal operatorer til rådighed. Operatorerne kan opdeles i tre grupper: De boolske operatorer, nærhedsoperatorerne og joker-operatorerne.

De boolske operatorer er OG, ELLER og IKKE. OG og ELLER giver mulighed for at opstille positive kriterier for, hvilke dokumenter der skal udvælges; IKKE giver mulighed for at opstille negative kriterier, dvs mulighed for at angive hvad der ikke skal udvælges. IKKE-operatoren udgør en risiko for eksplicit, men utilsigtet, at udelukke relevante dokumenter. Vi vurderer, at den ikke vil blive brugt ret meget: Søgning er en proces, hvor brugeren gennem formulering og reformulering af forespørgsler forsøger at finde de

dokumenter, der er relevante for hans/hendes informationsbehov. I denne proces er brugeren søgende fremfor sikker og derfor ikke tilbøjelig til eksplicit at udelukke nogle ord. Denne vurdering underbygges af, at hverken udvidet boolsk søgning eller begrebsbaseret fremfinding har IKKE-operatoren med. Vi afgrænser os derfor fra IKKE-operatoren.

Edb-Karnov omfatter således to boolske operatorer, OG og ELLER. For at gøre forespørgslerne utvetydige skal det defineres hvilken af de to operatorer, der binder stærkest. I matematik og programmeringssprog binder OG normalt stærkest - x ELLER y OG z er normalt det samme som x ELLER (y OG z). I søgesystemer bruges ELLER-operatoren imidlertid ofte til at angive synonymer; x ELLER y OG z har således betydningen (mindst) et af synonymerne x og y og derudover ordet z. Dette problem kunne løses ved brug af parenteser; vi vælger imidlertid blot at lade ELLER binde stærkere end OG. Dette valg genfindes i LEXIS (Harrison 1981).

Enhver OG-operator udføres indenfor en eller anden kontekst; men konteksten er ofte underforstået - OG betyder 'i samme dokument som'. Med *nærhedsoperatorer* er det muligt eksplicit at angive den kontekst, en OG-operator skal gælde indenfor. Nærhedsoperatorer bruges ud fra en antagelse om, at der er en tættere semantisk forbindelse mellem to ord, der forekommer i fx samme sætning, end mellem to ord, der blot forekommer i samme dokument. Det er meget let at finde såvel eksempler, der støtter denne antagelse, som eksempler, der modsiger den. Vi finder det vigtigt at fastholde, at nærhedsoperatorer - ligesom de boolske operatorer - blot afgør, hvorvidt ord forekommer sammen. Det udelukker naturligvis ikke, at der også er en semantisk forbindelse mellem ordene; men den udtaler søge-systemet sig ikke om.

Nærhedsoperatorer har vundet stor udbredelse ikke blot i litteraturen, men også i de kommercielt tilgængelige fuldtekstsøgesystemer. Der findes to forskellige typer af nærhedsoperatorer: Den første giver mulighed for at specificere nærheden i form af det antal ord, der maksimalt må være mellem de to søgeord; her spiller tekstens opbygning i sætninger, afsnit osv ingen rolle. Den anden følger tekstens opbygning og giver anledning til nærhedsoperatorer som fx 'indenfor samme afsnit'.

Efter vores mening afhænger nærhedsoperatorers anvendelighed af, om de refererer til en kontekst med en vis selvstændig betydning. Hvis konteksten har en vis selvstændig betydning, er der en rimelig chance for, at den indeholder alle de termer, der er centrale for beskrivelsen af hele dokumentets indhold. Efter vores mening er dét en betingelse for, at nærhedsoperatorer fungerer efter hensigten: Hvis denne betingelse ikke er opfyldt, vil der efter al sandsynlighed være en del relevante dokumenter, der findes frem ved brug af OG-operatoren, men ikke ved brug af en nærhedsoperator. Der refereres ikke til en kontekst med en vis selvstændig betydning i de tilfælde, hvor konteksten specificeres som det antal ord, der maksimalt må være mellem to søgeord. For de nærhedsoperatorer, der følger tekstens opbygning, er der i lidt højere grad tale om en kontekst med en vis selvstændig betydning. Efter vores mening bør valget af nærhedsoperatorer imidlertid afhænge af det specifikke område, søgesystemet er rettet mod.

Noterne i Edb-Karnov er grupperet ud fra hvilken lovtekst, de er knyttet til; men indenfor den enkelte note er der ingen kontekst med en vis selvstændig betydning. Ved søgning i noterne har OG-operatoren derfor altid betydningen 'indenfor samme note'. I lovteksterne er der imidlertid to kontekster, der har en vis selvstændig betydning: Paragraffer og stykker. Både paragraffer og stykker har yderligere den fordel, at nærhedsoperatorerne 'indenfor samme paragraf' og 'indenfor samme stykke' refererer til enheder, som jurister tænker i og er fortrolige med. Det står i kontrast til fx nærhedsoperatoren 'indenfor samme afsnit', der ville introducere en enhed, som ingen var vant til at opfatte lovtekster, som opdelt i. Derudover er nærhedsoperatoren NABO

relevant, da den giver mulighed for at søge på vendinger - og juridisk sprogbrug er rigt på vendinger og faste ordkonstellationer. For at reducere arbejdet med implementeringen af søgefaciliteterne afgrænser vi os til én nærheds-operator udover den uomgængelige 'indenfor samme lovtekst'. I lovteksterne stilles nærhedsoperatoren 'indenfor samme paragraf' således også til rådighed.

Nærhedsoperatorer angives normalt ved at erstatte OG-operatoren med en operator, der angiver den valgte nærhedsoperator. Ex: I Retsinformation betyder 'uddannelse s barsel', at ordene 'uddannelse' og 'barsel' skal forekomme i samme sætning. Det giver mulighed for forespørgsler, der kombinerer flere forskellige nærhedsoperatorer. Det taler for denne løsning, at fx Tenopir & Ro (1990) anbefaler meget, hyppig og meget differentieret brug af nærhedsoperatorer. Alternativt kan forespørgsler, der kombinerer flere forskellige nærhedsoperatorer, udelukkes, og nærhedsoperatoren angives ved et globalt valg, der gælder hele forespørgslen. I så fald bruges OG-operatoren i forespørgslerne, mens valget af nærhedsoperator sker udenfor selve forespørgslen og fastlægger konteksten for alle forespørgslens OG-operatorerne. Denne mulighed er lettere at implementere og forenkler formuleringen af forespørgslerne.

Bing (1981) bemærker, at undersøgelser af brugeres søgeadfærd viser, at de fleste holder sig til meget simple forespørgsler. Han mener, mange af forespørgslerne blot består af et enkelt søgeord. Bings artikel er ti år gammel; men efter vores mening stadig aktuel. Heroverfor står Tenopir & Ro's anbefaling, der forudsætter klarhed omkring, hvordan fuldtekstsøgning foregår. Det er vores vurdering, at kun de færreste brugere har denne klarhed. Derudover er jurister afhængige af et meget højt *recall*. Da snævrere nærhedsoperatorer udelukkende bruges i håb om at reducere antallet af irrelevante tekster blandt de fremfundne - de finder ingen nye tekster frem - må jurister forventes at være forsigtige med at vælge snævre nærhedsoperatorer. Specielt må jurister være interesserede i let at kunne finde ud af, hvor stor forskel valget af nærhedsoperator gør. På denne baggrund vurderer vi, at de har større behov for meget let at kunne gentage en forespørgsel med en anden nærhedsoperator end for at kunne bruge flere forskellige nærhedoperatorer i samme forespørgsel. Vi vælger derfor at implementere nærhedsoperatoren som et globalt valg, der gælder alle forespørgslens OGoperatorer.

Joker-operatorer giver mulighed for at angive, at her skal stå et vilkårligt bogstav eller en vilkårlig følge af bogstaver. Joker-operatorer bruges til at gøre forespørgslerne ufølsomme overfor ordenes bøjningsformer, overfor sammensætning af flere ord til ét sammensat ord og lignende. De anvendes ligeledes til at gøre forspørgslerne ufølsomme overfor variationer i brugen af store og små bogstaver. Da vi kun opererer med små bogstaver i søgeordstabellerne, er denne anvendelse imidlertid irrelevant i forbindelse med Edb-Karnov. Joker-operatorer er meget væsentlige i forbindelse med fuldtekstsøgning, da de bidrager til at reducere springet fra begreb til term. Implementering af joker-operatorer i Edb-Karnov er let, da Oracle umiddelbart giver denne mulighed. I forespørgslerne kan ét vilkårligt bogstav således angives med '_', og en vilkårlig følge af bogstaver med '%'. Der er ingen begrænsninger på placeringen eller antallet af joker-operatorerne. Ex: Såvel 'b_rn%' som '%miljø%' og 'virksomhed%karnov' er tilladte søgeord. I Edb-Karnov indgår joker-operatorerne direkte i forespørgslerne.

En anden mulighed var at bruge joker-operatorerne til at søge i søgeordstabellerne. En sådan søgning ville resultere i en liste med ord, der efterfølgende kunne føjes til forespørgslen eller droppes. Derved kunne det undgåes, at joker-operatorerne førte til udvidelser af forespørgslerne med utilsigtede søgeord. Tenopir & Ro (1990) anbefaler, at fuldtekstsøgesystemer giver brugerne en eller anden mulighed for at søge i søgeordstabellerne; det er en god støtte i valget af søgeord. Vi er enige i, at brugerne skal tilbydes støtte i valget af søgeord; men vi mener, den bedste måde at gøre det på er ved

hjælp af en tesaurus. Vi tror, søgning i søgeordstabellerne ofte tilbydes, fordi det er meget simpelt at implementere, og fordi søgesystemet ikke har en on-line tesaurus.

I forbindelse med forespørgslerne skal der også angives nogle parametre, der fastlægger søgningens omfang. Det er vigtigt, at parametrenes aktuelle værdier altid er synlige, og at de er lette at ændre. Af den grund er indstillingen og visningen af parametrene placeret fast på søge-skærmbilledet (se figur 4.4) - alternativet er et 'pop up'-vindue. For at fastlægge en søgnings omfang skal brugeren på den ene side bestemme hvilke typer tekst, søgningen skal omfatte. Her er mulighederne lovtekster og/eller noter. På den anden side skal det fastlægges, hvilke lovtekster/noter der skal søges i. Her skelner vi mellem to tilfælde: Brugeren kan søge efter tekster - lovtekster og/eller lovteksternes noter - blandt alle teksterne i Edb-Karnov, eller brugeren kan søge efter specifikke paragraffer og/eller noter indenfor en bestemt tekst.

Ovenfor er de to parametre beskrevet, som de kan implementeres ud fra grundstammen i datamodellen; senere tilføjes endnu en valgmulighed til begge parametrene. I afsnit 4.7 føjes teksttypen notater til lovtekster og noter, således at søgningerne kan foregå i tre forskellige typer tekst. Med hensyn til den anden parameter er muligheden for at afgrænse søgningen til én eller alle tekster ikke dækkende; der er også behov for at kunne afgrænse søgningen til netop den gruppe tekster, brugeren finder relevante. Denne mulighed er emnet for afsnit 4.5. Vi afslutter behandlingen af forespørgslerne med et eksempel på en af de SQL-sætninger, forespørgslerne giver anledning til:

```
SELECT DISTINCT L.ART, L.AAR, L.NUMMER, L.NAVN
FROM LOVOPL L, LOVSOEGEORD S
WHERE (S.SOEGEORD LIKE 'barsel%')
AND S.LOVID = L.LOVID
INTERSECT
SELECT DISTINCT L.ART, L.AAR, L.NUMMER, L.NAVN
FROM LOVOPL L, LOVSOEGEORD S
WHERE (S.SOEGEORD LIKE 'fyring%'
OR S.SOEGEORD LIKE 'afskedigelse%')
AND S.LOVID = L.LOVID
ORDER BY 2, 3
```

Figur 4.5. Eksempel på en af de SQL-sætninger, Edb-Karnov genererer ud fra brugernes forespørgsler. Forespørgslen, der har givet anledning til denne SQL-sætning, er: 'barsel% og fyring% eller afskedigelse%'. Nærhedsoperatoren er 'indenfor samme lovtekst'; omfanget er alle lovtekster.

Præsentationen - resultatet af søgningen

En søgning giver et resultat i form af nogle tekster, der matcher forespørgslen. Præsentationen af disse tekster sker i Edb-Karnov i to omgange: Først vises en oversigt over samtlige fremfundne tekster, derefter kan brugeren vælge at se den fulde tekst for en af teksterne. Efter at have undersøgt denne tekst kan brugeren vende tilbage til oversigten og vælge at se den fulde tekst for en anden af de fremfundne tekster. Oversigten hænger direkte sammen med forespørgslen og vil ofte give anledning til en reformulering af den. Det ville derfor være en fordel, hvis oversigten og forespørgslen var synlige samtidig. På grund af Hypercard-kortenes størrelse er vi imidlertid nødt til at vise enten oversigten eller forespørgslen.

Oversigten over de fremfundne tekster skal identificere de enkelte tekster. For lovteksterne sker det ved angivelse af art, år, nummer og navn; for noterne angives, at der

er tale om en note til den og den lovtekst. Når der søges indenfor en enkelt tekst, er det overflødigt at identificere teksten. Her angives blot de fundne paragraffer og/eller noter. Udover disse referenceoplysninger kan det overvejes, om oversigten skal indeholde de passager, søgeordene forekommer i. Da en forespørgsel normalt indeholder flere søgeord, og hvert søgeord ofte forekommer flere gange i en tekst, vil passagepræsentation give anledning til en voldsom udvidelse af oversigtens omfang. Det mener vi ikke er hensigtsmæssigt, dels fordi det vil gøre oversigten væsentligt mindre overskuelig, dels fordi den fulde tekst er så let tilgængelig. Et ofte benyttet alternativ til passagepræsentation er at fremhæve søgeordene i teksten, så de er lette at få øje på ved en hurtig gennembladring af den. Vi finder denne mulighed tiltalende og væsentligt bedre end passagepræsentation. Vi har imidlertid ikke implementeret den, da det i Hypercard er besværligt at fremhæve enkelte ord i en tekst.

Mange af teksterne i Edb-Karnov er ret lange. Vi formoder derfor, at brugeren efter en søgning, der har fremfundet hele tekster, ofte vil være interesseret i at gentage søgningen indenfor en af de fremfundne tekster. Ved søgning indenfor én tekst fremfindes de specifikke paragraffer og/eller noter, der opfylder forespørgslen. Oversigten over disse paragraffer og/eller noter kan enten erstatte eller støtte gennembladring af hele teksten. Denne gentagelse af en søgning indenfor en af de fremfundne tekster kan i Edb-Karnov udføres blot ved at klikke på den pågældende tekst i oversigten; derefter spørges brugeren, om han/hun vil gentage søgningen indenfor den valgte tekst eller se selve teksten.

I dette og det foregående afsnit er de grundliggende dele af Edb-Karnov diskuteret og beskrevet. På dette tidspunkt er der mulighed for boolsk søgning med joker-operatorer og en enkelt nærhedsoperator. Søgningerne kan omfatte én eller alle lovtekster og/eller noter. I de følgende afsnit udvides Edb-Karnov med tre væsentligt mere avancerede faciliteter. De to første sigter på at give bedre muligheder for at søge i de tekster, systemet indeholder; den tredje giver mulighed for at udvide Edb-Karnov med nye tekster.

4.5 Dynamisk emneregister

I dette afsnit behandles Edb-Karnovs emneregister, der er rettet mod valget af de tekster, søgningen skal foregå i. Emneregistrets vigtigste funktion er at reducere antallet af irrelevante dokumenter, der matcher forespørgslen. Det gøres ved at begrænse søgningens omfang til en del af søgesystemets dokumenter. Emneregistret er således Edb-Karnovs facilitet til det, vi i afsnit 2.5 kaldte opdeling af databasen i klynger. Fordelen ved at opdele databasen i klynger er, at der opnåes højere *precision*, uden at *recall* påvirkes nævneværdigt. Eller sagt på en anden måde: Søgningerne kan gøres bredere, uden at *precision* bliver uacceptabelt lav.

Edb-Karnovs emneregister skal opfylde to formål: For det første skal det - som allerede nævnt - give mulighed for at afgrænse søgningerne til en udvalgt del af Edb-Karnovs dokumenter. For det andet skal emneregistret give et overblik over strukturen i lovgivningssystemet og de enkelte lovteksters placering i denne struktur. For at opfylde dette formål mener vi, brugerne skal have mulighed for at ændre i emneregistret - emneregistret skal kunne udvikles over tid. Herved får brugerne mulighed for at lade emneregistret afspejle præcis den opdeling af lovteksterne, de selv benytter og har behov for.

I dette afsnit vil vi først diskutere ideen bag Edb-Karnovs emneregister. Derefter behandles grundemneregistret. Efter det diskuteres og beskrives emneregistrets tre funktioner: Navigation i emneregistret, brugen af emneregistret i søgningerne og redigering af emne-registret. Endelig afsluttes afsnittet med en beskrivelse af de tilføjelser

til datamodellen, emneregistret giver anledning til.

Ideen bag Edb-Karnovs dynamiske emneregister

Edb-Karnov skal støtte fagfolk i deres arbejde. For at gøre det, må opdelingen af teksterne i Edb-Karnov svare til den, brugerens sag kræver. Ellers vil systemet være tungt at arbejde med og forstyrre brugeren i sagsbehandlingen. For at emneregistret kan afspejle den opdeling af teksterne, som brugeren finder mest hensigtsmæssig, må brugeren efter vores mening have mulighed for løbende at ændre i det. Der ligger to ideer bag dynamikken i Edb-Karnovs emneregister.

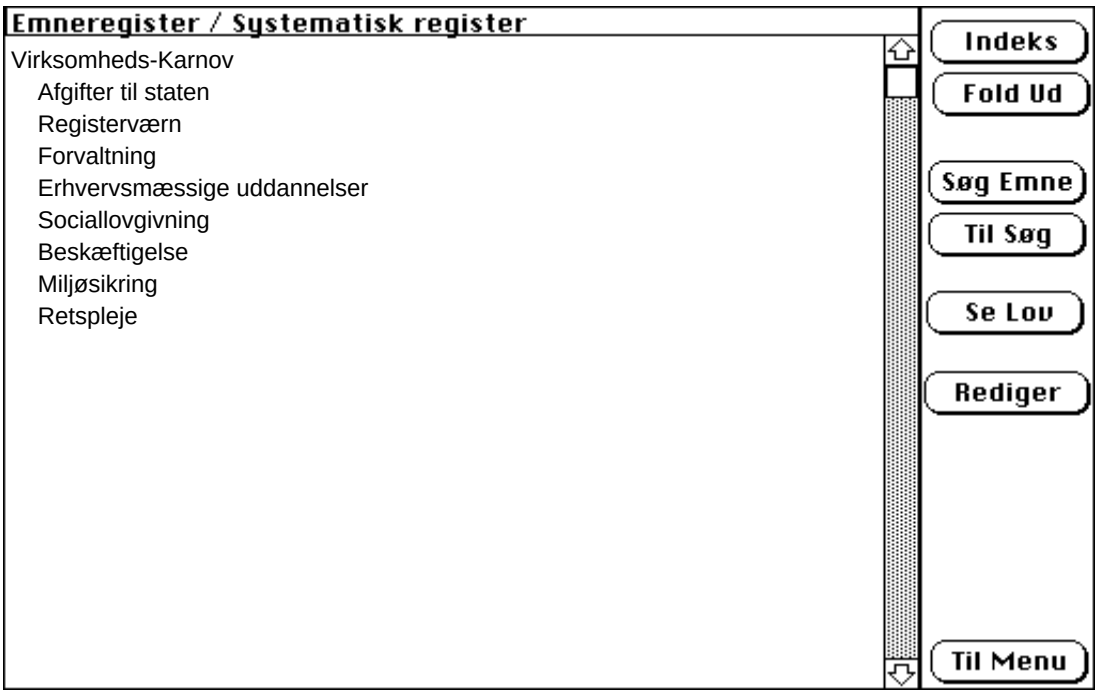
For det første skal emneregistret være et netværk, ikke blot et hierarki. I et hierarki tilhører hver tekst netop ét af registrets emner. Denne opdeling er den mest udbredte og svarer til en bogs indholdsfortegnelse eller Retsinformations opdeling af dokumenterne i disjunkte baser. Von Eyben (1989) påpeger, at en hierarkisk systematik ikke svarer til virkeligheden, når det gælder opdeling af lovtekster. En lovtekst kan fx være relevant indenfor både landbrugsområdet og miljøområdet; men i et hierarki skal den placeres indenfor ét emne. Hvis der i stedet anvendes en netværkssystematik, kan en lovtekst tilhøre flere forskellige emner. En væsentlig fordel ved netværk er, at der kan ændres i opdelingen af teksterne, uden at den gamle opdeling går tabt. Det er jo ikke nødvendigt at fjerne en lovtekst fra det emne, den i øjeblikket er placeret i, før den kan placeres i et andet. Vi mener, en netværkssystematik giver juristen en bedre støtte i arbejdet, og baserer derfor Edb-Karnovs emneregister på en netværkssystematik.

For det andet skal emneregistret give brugeren mulighed for at oprette sine egne sagsorienterede emneregistre. En jurist har forskellige typer sager, der hver lægger deres perspektiv på lovteksterne. Hvis brugeren behandler mange sager af samme type, ser vi en fordel i, at alle de lovtekster, der er relevante for denne type sager, kan samles i et særskilt emneregister. Derefter kan søgninger i tilknytning til sådanne typiske sager umiddelbart afgrænses til de lovtekster, der tilhører det pågældende emneregister. Dorthe la Cour synes godt om muligheden for at oprette et emne, der svarer til en sag eller sagstype. Den kan hjælpe hende med at huske, hvilke lovtekster der var relevante for sagen, og dermed lette behandlingen af en lignende sag på et senere tidspunkt. Hun påpegede imidlertid også, at afgrænsning af søgningerne til en udvalgt del af lovteksterne kan betyde, at et vigtigt dokument ikke findes. Dorthe la Cour nævnte et grelt, men dog også usædvanligt, eksempel: Man har ændret i sygedagpengeloven; men ændringen er foretaget i lov om socialstyrelser, hvor den er opført sammen med oplysningerne om, hvilke paragraffer der erstattes/ændres med den nye udgave af lov om socialstyrelser.

I et netværk er det muligt at have forskellige 'syn' på de samme tekster, så de to ideer er knyttet tæt sammen. Med et netværk er det naturlige udgangspunkt for opdelingen, at teksterne kan grupperes helt frit, således at et emne kan bestå udelukkende af noter, af bekendtgørelser indenfor et givet område, af noter og lovtekster osv. Der er imidlertid en tæt sammenhæng mellem de forskellige tekster; en note er knyttet til en lovtekst, en bekendtgørelse er knyttet til lov osv. Den juridiske metode (se afsnit 3.1) tager netop udgangspunkt i denne opbygning af lovgivningssystemet. Vi mener, det bør være muligt at placere bekendtgørelser i emneregistret uafhængigt af den lov, de er knyttet til; men vi ser ikke noget behov for at kunne placere noter uafhængigt af lovteksterne. I Edb-Karnovs emneregister systematiseres teksterne derfor ud fra lovteksterne. Enhver lovtekst kan placeres efter brugerens ønske; noterne følger altid de lovtekster, de er knyttet til.

Emneregistret giver anledning til et nyt skærmbillede (se figur 4.6). Dette skærmbillede er tæt knyttet til søgeskærmbillede. Muligheden for at afgrænse søgningen til en udvalgt del teksterne kan indlede søgningen. Under formuleringen af forespørgslen har brugeren

imidlertid også mulighed for at gå ind i emneregistret og fastlægge søgningens omfang.



Figur 4.6. Emneregistrets skærmbillede. Til venstre viser figuren emneregistrets opdeling (her det øverste niveau). Til højre ses funktionerne: Indeks og Fold Ud der bruges ved navigation i emneregistret, Søg Emne og Til Søg der skifter til søge-skærmbilledet, Se Lov der viser den valgte lovtekst på lovtekst-skærmbilledet, Rediger der er redigeringsfunktionen og Til Menu der skifter til menuskærmbilledet.

Grundemneregistret

Edb-Karnovs emneregister kan betragtes som bestående af to sammenvævede dele: Et grundemneregister og de ændringer, brugeren har foretaget. Vi ser to væsensforskellige muligheder i valget af grundemneregister. Den ene tager udgangspunkt i informationsleverandørerne (ministerierne). Den anden mulighed er at systematisere teksterne ud fra, hvad der er hensigtsmæssigt i brugssituationen. Da Edb-Karnov skal støtte jurister i deres arbejde, vælger vi den anden mulighed. Emneregistret i Virksomheds-Karnov er rettet mod brugssituationen og omfatter de tekster, vi arbejder med. Vi vil derfor bruge emneregistret i Virksomheds-Karnov som grundemneregister i Edb-Karnov.

Emneregistret i Virksomheds-Karnov svarer til en indholdsfortegnelse. De lovtekster, vi arbejder med, er i registret systematiseret i otte overordnede emner (retsområder). Hvert af disse emner er igen opdelt i underemner, der igen kan være yderligere opdelt. Emneregistret i Virksomheds-Karnov er hierarkisk opbygget, så Edb-Karnovs grundemneregister bliver hierarkisk. Brugerne kan naurligvis stadig udnytte, at Edb-Karnovs emneregister giver mulighed for at organisere lovteksterne i et netværk.

Emneregistret i Virksomheds-Karnov kan ses som et 'syn' på lovteksterne i Karnovs Lovsamling, mens det systematiske register i Karnovs Lovsamling kan ses som et andet. Ex: I Virksomheds-Karnov består emnet 'Sociallovgivning' af to love: Lov 1989 nr 852 og lbkg 1987 nr 450. I Karnovs Lovsamling indeholder det samme emne mange flere lovtekster, og det er opdelt i to underemner: 'A. Almindelige regler om social bistand' og 'B. Andre sociale ydelser' (se figur 4.7). Vi ser dette som et eksempel på, at der er behov for flere forskellige grundemneregistre, fx et generelt og et specifikt for

virksomhedsjuraen. Denne mulighed ligger i direkte forlængelse af vores idé om, at brugeren kunne oprette sine egne sagsorienterede emneregistre. I Edb-Karnov afgrænser vi os til at medtage ét grundemneregister; men det er umiddelbart muligt at lægge andre ind ved siden af. Hvis der lægges flere grundemneregistre ind, vil det øverste niveau i det samlede emneregister udgøre en oversigt over de forskellige 'syn' på lovteksterne.

Sociallovgivning

Lov om dagpenge ved sygdom eller fødsel, **L 1989** 852
Lov om arbejdsskadeforsikring, **lbkg 1987** 450 ændret ved **L 1987** 871, **L 1988** 743, **L 1989** 196 og **L 1989** 389

figur 4.7 (a)

7. Sociallovgivning
A. Almindeligeregler om social bistand

.
.
.

B. Andre sociale ydelser
1. Børnetilskud

.
.
.

4. Dagpenge ved sygdom eller fødsel
Dagpenge ved sygdom eller fødsel, **L 1989** 852

5. Tilskadekomne
Arbejdsskadeforskikring, **lbkg 1987** 450 ændret ved **L 1987** 871, **L 1988** 743, **L 1989** 196 og **L 1989** 389

.
.
.

6. Tillægspension

.
.
.

figur 4.7 (b)

Figur 4.7. Et udsnit af emneregistret fra (a) Virksomheds-Karnov og (b) Karnovs Lovsamling. Figuren viser, at de to emneregistre udgør to forskellige 'syn' på lovteksterne. De to love fra Virksomheds-Karnov er i Karnovs Lovsamling placeret under henholdsvis '4. Dagpenge ved sygdom eller fødsel' og '5. Tilskadekomne'.

Navigation

Mulighederne for at navigere i emneregistret hænger nøje sammen med, hvordan emneregistret præsenteres. En mulighed er at præsentere emneregistret som én lang tekst, der kan bladres i. Denne mulighed giver ikke noget særlig godt overblik over systematikken, da alle emner på samme niveau ikke vises samlet. Det skyldes skærmens begrænsede størrelse og, at alle underemner vises mellem emner på samme niveau. En helt anden mulighed er at præsentere registret grafisk som et træ med mulighed for at hoppe fra knude til knude. Denne mulighed er efter vores mening for uoverskuelig, da der er mange emner, som skal forbindes.

En tredje mulighed, der til en vis grad kombinerer de to ovennævnte, er at præsentere emneregistret som en tekst, hvor hvert emne kan foldes ud og pakkes sammen. Som

udgangspunkt vises blot emneregistrets navn. Ved at klikke på navnet foldes registret ud, dvs alle emnerne på registrets øverste niveau præsenteres. Hvert emne præsenteres på en linie for sig og indrykket under registrets navn. Ved at klikke på et eller flere af disse emner, vil de også blive foldet ud (se figur 4.8). Når brugeren ikke længere ønsker, at et emne er foldet ud, kan det pakkes sammen igen. På denne måde har brugeren fuld kontrol over, hvilke dele af emneregistret der vises, og hvilken detaljeringsgrad de vises med.

```
Virksomheds-Karnov
Afgifter til staten
Registerværn
Forvaltning
Erhvervsmæssige uddannelser
Sociallovgivning
Beskæftigelse
  Arbejdsret
  Funktionærer og medhjælpere
  Ferie
    Lov om ferie, lbkg 1984....
    - Bekendtgørelse
  Arbejdsmiljø
Miljøsikring
Retspleje
```

Figur 4.8. Emneregister med mulighed for at folde emner ud og pakke dem sammen igen. Selve registret samt emnerne 'Beskæftigelse', der omfatter fire underemner, og 'Ferie', der omfatter to lovtekster, er foldet ud. Som det næste kunne fx 'Forvaltning' foldes ud eller 'Ferie' pakkes sammen.

I Edb-Karnov vil vi afgrænse os til en forenklet udgave af ovenstående forslag. Afgrænsningen består i, at der kun kan være ét udfoldet emne ad gangen; derudover vises dette emnes overemner hele vejen op til registrets navn (se figur 4.9). I Edb-Karnov er det således ikke muligt at have både 'Beskæftigelse' og 'Ferie' foldet ud på én gang. Overblikket, over hvor det udfoldede emne hører til i systematikken, søges imidlertid bevaret ved at vise overemnerne helt op til registrets øverste niveau.

```
Virksomheds-Karnov
Beskæftigelse
  Ferie
    Lov om ferie, lbkg 1984....
    - Bekendtgørelse
```

Figur 4.9. Edb-Karnovs emneregister. Figuren viser emnet 'Ferie' udfoldet, og at det er et underemne 'Beskæftigelse'.

Navigering i emneregistret foregår ved at markere et emne og klikke på Fold Ud. Når et emne foldes ud, bliver eventuelle udfoldede emner på underliggende niveauer pakket sammen, da det kun er muligt at have ét udfoldet emne ad gangen. Navigation indenfor et emneregister kan således foregå udelukkende ved brug af Fold Ud. Hvis brugeren har installeret flere forskellige emneregistre, skal der også være mulighed for at skifte mellem dem. Det sker ved at klikke på Indeks. Når denne funktion aktiveres, pakkes et eventuelt udfoldet emne sammen, og det samlede emneregisters øverste niveau vises, dvs der vises en oversigt over de installerede emneregistre.

Søgning

Emneregister i Edb-Karnov skal bruges til at begrænse den del af Edb-Karnovs tekster, den følgende søgning skal omfatte. Formålet med denne specifikation af, hvilke tekster der skal søges i, er at reducere antallet af fremfundne, men irrelevante, tekster. Det skal således være muligt at udvælge de tekster, en søgning skal omfatte. Den optimale løsning er at give brugeren mulighed for at plukke de emner og specifikke lovtekster, søgningen skal afgrænses til. Efterhånden, som emner og/eller specifikke lovtekster plukkes, kan de placeres i en pulje. Når brugeren afslutter plukningen, vil denne pulje angive den del af Edb-Karnovs tekster, der skal søges i.

Vi afgrænser os til en skrabet udgave af denne løsning. I Edb-Karnov er det kun bliver muligt at plukke ét emne eller én lovtekst. Søgningerne kan således afgrænses til ét emne eller én specifik lovtekst. Afgrænsningen af søgningen sker ved, at brugeren navigerer sig hen til og markerer det ønskede emne/den ønskede lovtekst, derefter klikkes på Søg Emne. Derved skiftes til søge-skærmbilledet, hvor søgningen kan startes. Søgningen foregår nu som beskrevet i afsnit 4.4, men omfatter kun de tekster, der er i puljen fra emneregistret.

Redigering

Baggrunden for at Edb-Karnov har et emneregister, er først og fremmest, at denne facilitet er relevant i forbindelse med systemets udvikling over tid. Betingelsen, for at systemet kan udvikle sig over tid, er, at brugeren kan foretage løbende ændringer. Denne redigeringsmulighed kan bruges til at justere grundemneregistret efter brugerens specifikke og skiftende behov, og den kan specielt bruges til at oprette sagsorienterede emneregistre. Det er derfor, vi kalder det et dynamisk emneregister.

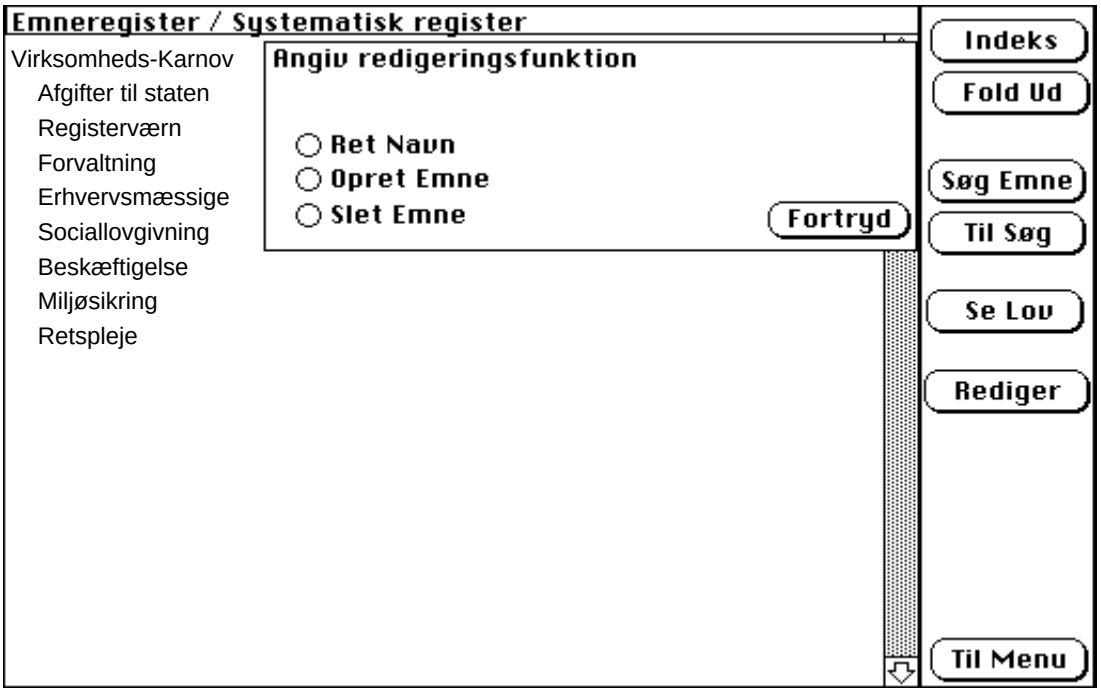
Redigeringsmulighederne og brugen af emneregistret i søgningerne kan enten adskilles eller integreres. Ved at adskille de to aktiviteter bliver det nemmere at styre og kontrollere emneregistrets udvikling. Da behovet for at ændre i emneregistret opstår i forbindelse med søgningerne mener vi imidlertid, de to aktiviteter skal integreres. Redigeringen skal indgå i brugen af Edb-Karnov som en naturlig og nærliggende mulighed, der står til rådighed så snart behovet for at ændre opstår. Hvis det ikke er tilfældet, tror vi ikke, redigeringsmulighederne vil blive udnyttet.

Edb-Karnovs redigeringsfunktioner kan opdeles i de, der gælder for emner, og de, der gælder for lovtekster. Et emne skal kunne tilføjes og slettes som underemne for et andet emne (se figur 4.10). Når et emne tilføjes, kan der enten være tale om indplacering af et allerede eksisterende emne eller om oprettelse af et nyt. Vi har samlet de to typer tilføjelser i én funktion. Hvis det skal være muligt at have flere emner med samme navn, er det nødvendigt med to separate funktioner. Da emneregistret er et netværk, betyder indplacering af et allerede eksisterende emne, at dette emne får mere end ét overemne. En ændring i det indplacerede emne vil således gælde både for emnets gamle og nye placering. En tredje form for tilføjelse er derfor relevant: Kopiering. Kopiering vil specielt være relevant, hvis brugeren arbejder med flere forskellige emneregistre. Vi afgrænser os til oprettelse og indplacering, kopiering vil ikke blive behandlet yderligere. Sletning af et emne betyder, at forbindelsen mellem emnet og det overemne, der vises på skærmen, slettes. Hvis emnet efter sletningen stadig er knyttet til andre emner, sker der ikke mere. Hvis det ikke er tilfældet, bliver selve emnet slettet.

Det kan være, at sprogbrugen ændrer sig, så brugeren ikke længere finder et emnes navn dækkende. Det er derfor muligt at ændre et emnes navn, uden at det iøvrigt berører emneregistret. En lignende funktion kunne være relevant til ændring af ledeteksterne for lovteksterne; den har vi imidlertid ikke implementeret.

For lovteksterne omfatter redigeringsfunktionerne indplacering af en lovtekst i et emne og sletning af en lovtekst fra et emne. Indplacering af en lovtekst foregår ved

udvælgelse fra en kronologisk oversigt over lovteksterne. En sletning består i at fjerne forbindelsen mellem lovteksten og det pågældende emne.



Figur 4.10. De tre redigeringsfunktioner for et emne: Ret Navn der ændrer det markerede emnes navn, Opret Emne der tilføjer et emne som underemne for det markerede emne og Slet Emne der sletter det markerede emne fra dette sted i emneregistret.

Datamodel

Vi sammenfatter afsnittet med en kort beskrivelse af de tilføjelser til datamodellen, emneregistret giver anledning til (se figur 4.11). De emner, emneregistret omfatter, bliver entydigt identificeret ved et emneid. Det gør det nemt at udvide emneregistret så flere emner kan have samme navn.

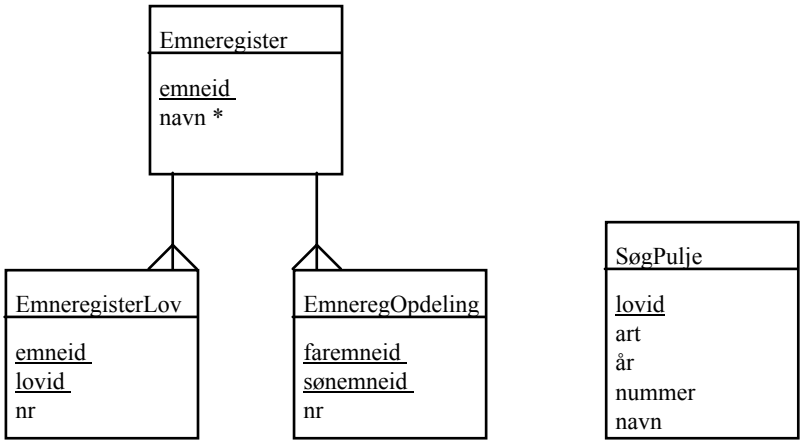
Alle emner beskrives ved emneid og navn i emneregistrets grundtabel *emneregister*. Opdelingen af emnerne giver anledning til tabellen *emneregisteropdeling*. Hvis et emne er underemne for et andet, vil de to tilhørende emneid'er være sammenknyttet i tabellen. Denne sammenknytning af emner giver reelt mulighed for et vilkårligt antal niveauer i emneregistret. Vi finder det hensigtsmæssigt altid at præsentere et emnes opdeling, som den er indtastet. Det betyder, at det er nødvendigt at nummerere alle de emner, der har fælles overemne. Endelig skal det være muligt at beskrive, hvilke lovtekster et emne omfatter. Det sker ved hjælp af tabellen *emneregisterlov*. For at kunne præsentere lovteksterne i den rækkefølge, de er indtastet i, bliver de nummereret.

Den ledetekst, der beskriver hver enkelt lovtekst i emneregistret, afhænger ikke af det emne, lovteksten indgår i; den følger lovteksten (beskriver den). Derfor er tabellen *lovoplysninger* udvidet med attributen beskrivelse, der indeholder ledeteksterne.

Kommunikationen mellem emneregistret og søgningen foregår ved hjælp af en pulje, der angiver de lovtekster, søgningen skal omfatte. Af hensyn til såvel søgningernes svartider som implementeringen af emneregistret har vi fundet det hensigtsmæssigt, at denne kommunikation foregår ved hjælp af en tabel, *søgpulje*. Det er en dynamisk tabel, der altid tømmes, før den næste gang bruges til at angive de tekster, en søgning skal omfatte. Da denne tabel allerede findes, vil det være ret let at udvide emneregistret med en mulighed for at plukke flere emner og/eller lovtekster til den pulje, søgningen skal

omfatte.

På grund af *søgpulje* kan vi opnå rimelige svartider og samtidig have *emneregisterlov* på tredje og fjerde normalform. Nu er *emneregisterlov* defineret sådan, at den kun knytter lovtekster til emnerne på emneregistrets nederste niveau. Hvis vi ikke brugte *søgpulje*, ville det være nødvendigt at udvide *emneregisterlov*, så den til hvert emne knyttede de lovtekster, som indgik i emnet selv eller et af dets underemner. Med *søgpulje* har vi opnået, at alle fire nye tabeller er på tredje såvel som fjerde normalform.



Figur 4.11. Datamodel for Edb-Karnovs emneregister. Hvert emne identificeres entydigt ved et emneid. Tabellernes primærnøgle er markeret ved understregning, alternative nøgler med stjerner.

4.6 Dynamisk tesaurus

I dette afsnit diskuteres, fastlægges og beskrives Edb-Karnovs tesaurus. Mens emneregistret er rettet mod valget af de tekster, søgningen skal omfatte, er tesaurussen rettet mod valget af søgeord. En af tesaurussens vigtige funktioner er at reducere det centrale problem ved fuldtekstsøgning: Der skal bruges præcis de ord i forespørgslerne, som forekommer i teksterne (Eddison & Batty 1988). Det søges gjort ved at etablere relationer mellem tesaurussens begreber, typiske relationer er bredere, snævrere, synonyme og relaterede begreber. Derudover er der som oftest mulighed for at knytte en forklarende eller definerende note til hvert ord. Vi behandlede tesaurusfaciliteter i afsnit 2.5. Her fremgik det, at tesaurusfaciliteter er sjældne i de kommercielt tilgængelige søgesystemer. Der forskes meget i tesaurusser; men de tesaurusser, der bruges, er næsten alle på bogform. Det fremgik endvidere, at ajourføringen af tesaurussen næsten altid betragtes som en opgave, der skal varetages centralt af en særligt sagkyndig person eller gruppe.

I det følgende vil vi først diskutere, hvorfor vi mener, det er utilstrækkeligt udelukkende at foretage ajourføringen centralt. Derefter beskrives de relationer, som Edb-Karnovs tesaurus indeholder, og så behandles grundtesaurussen. Efter det diskuteres og beskrives tesaurussens tre funktioner: Navigation, støtte af såvel fuldtekstsøgning som nøgleordsbaseret søgning og redigering. Endelig afsluttes afsnittet med en beskrivelse af de tilføjelser til datamodellen, som tesaurussen giver anledning til.

Ideen bag Edb-Karnovs dynamiske tesaurus

Grundtanken i Edb-Karnov er, at det skal fungere som et redskab for fagfolk. Specielt skal tesaurussen tjene tre formål: For det første skal den tilbyde støtte i valget af søgeord. For

det andet skal den give mulighed for at supplere søgningerne ved tilføjelse af fx relaterede eller snævrere begreber. For det tredje skal den tilbyde en enkel form for nøgleordsbaseret søgning. For at tesaurussen kan leve op til det, må brugeren kunne ajourføre den under sit daglige arbejde. Vi mener ikke, vores idé om udvikling over tid kan honoreres, hvis ajourføringen udelukkende foretages centralt. Begrundelsen for dette er først og fremmest, at ajourføringen skal ske løbende og så snart, brugeren erkender behovet. Ellers vil systemet virke forstyrrende i brugerens arbejde: Hver gang systemet bruges skal brugeren huske eksplicit at højde for de variationer i sprogbrug og lignende, der ikke dækkes af tesaurussen. Det er et såvel enerverende som irriterende stykke arbejde med mange muligheder for fejl og forglemmelser.

En ændring af tesaurussen er velbegrundet, hvis og når brugeren finder den ønskelig. For at opfylde de tre ovennævnte formål over tid skal tesaurussen altså være dynamisk: Tesaurussen er ikke blot en statisk størrelse, der kan støtte formuleringen af forespørgslerne; formuleringen af forespørgslerne giver også anledning til ændringer og ajourføringer af tesaurussen. Vi søger at give tesaurussen den nødvendige dynamik ved hjælp af en redigeringsfunktion, som er integreret med tesaurussens øvrige funktioner. Dynamikken kan ikke blot udnyttes af den enkelte bruger; udbyderen af Edb-Karnov kan også have glæde af den. Når udbyderen skal ajourføre tesaurussen, vil det med fordel kunne gøres i samarbejde med nogle udvalgte brugere. Her giver Edb-Karnovs dynamiske tesaurus mulighed for at ændringer, der er foretaget i brugssituationen, inddrages i ajourføringen.

Edb-Karnov skal endvidere indeholde en facilitet, der muliggør en enkel form for nøgleordsbaseret søgning. Baggrunden for dette er, at fuldtekstsøgning og nøgleordsbaseret søgning støtter forskellige søgestrategier - i afsnit 3.4 talte vi om henholdsvis en *bottom-up* og en *top-down* søgestrategi. Disse to søgestrategier er væsensforskellige og begge meget vigtige; et godt søgesystem kan derfor ikke nøjes med at støtte den ene. Vi begrænser os til at medtage en facilitet til en enkel form for nøgleordsbaseret søgning og placerer den i tesaurussen. Denne placering er usædvanlig - vi har ikke set den beskrevet andre steder - og diskuteres senere.

Relationerne i tesaurussen

Edb-Karnovs tesaurus omfatter fem relationer. De fire er standardrelationer; den femte er grundlaget for Edb-Karnovs facilitet til nøgleordsbaseret søgning. De fire standardrelationer, vi har valgt, er efter vores mening de centrale og omfatter da også de tre, der går igen i litteraturen (bredere, snævrere og relaterede begreber). Vi kunne have tilføjet flere relationer; men de fem er fuldt tilstrækkeligt til at illustrere, hvordan en tesaurus kan bruges i forbindelse med fuldtekstsøgning, og hvordan den kan gøres dynamisk. De fem relationer i Edb-Karnovs tesaurus er:

Overbegreber. Denne relation refererer til en niveaudeling af begreberne, idet den angiver de begreber, der ligger ét niveau over et givet begreb. Relationen er en generalisering af den BT-relation ('Broader Than'-relation), der er beskrevet i afsnit 2.5. BT-relationen tillader kun ét overbegreb til hvert begreb og giver dermed kun mulighed for at opbygge et hierarki i form af en træstruktur. Der er to grunde til, at Edb-Karnovs overbegreber-relation ser ud, som den gør: For det første mener vi, at det er af stor værdi lokalt at kunne placere begreber over eller under hinanden. For det andet finder vi, at hierarkier er alt for restriktive til at beskrive de globale sammenhænge mellem begreberne. Med Edb-karnovs overbegreber-relation er det muligt at opbygge et netværk, hvor hvert begreb har flere overbegreber, samtidig med at vi fastholder muligheden for lokalt at placere begreberne over eller under hinanden.

Underbegreber. Denne relation angiver de begreber, der ligger ét niveau under et

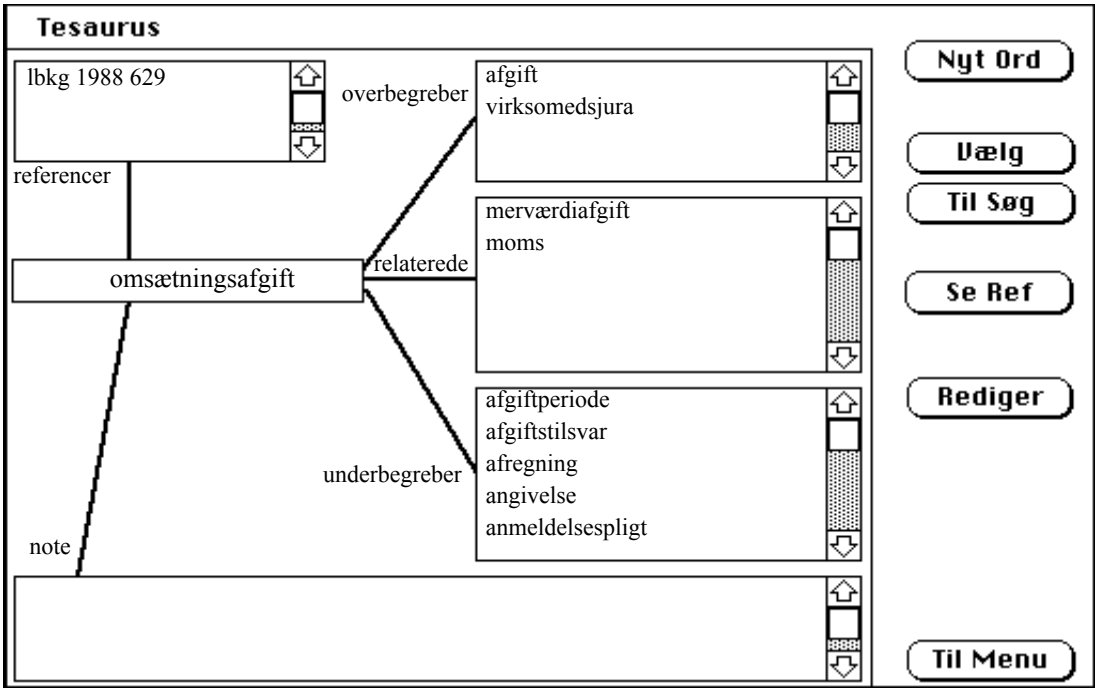
givet begreb. Relationen svarer til overbegreber-relationen bortset fra, at den går et niveau nedad i stedet for opad. I forhold til de tesaurusrelationer, der er beskrevet i afsnit 2.5, er underbegreber-relationen identisk med NT-relationen ('Narrower Than'-relationen).

Relaterede begreber. Denne relation omfatter såvel synonyme begreber som næsten synonyme og andre beslægtede begreber. Vi har defineret denne relation meget pragmatisk: Den knytter et givet begreb til alle de relevante begreber, der hverken kan rubriceres som over- eller underbegreber. Denne relation omfatter således både synonymer, meningslignende ord (ord der næsten er synonymer og ofte betragtes som sådan) og andre relaterede begreber på samme niveau i hierarkiet.

Note. Denne relation giver mulighed for at knytte en beskrivelse, definition eller lignende til et begreb. Denne relation er værdifuld i sig selv, fordi den giver brugeren mulighed for at knytte netop den bemærkning til et begreb, som han/hun har brug for. Relationen er imidlertid også grundlaget for adskillelsen af et homografs forskellige betydninger. Selvom de staves ens, skal alle en homografs forskellige betydninger naturligvis kunne indgå i tesaurussen uden at blive blandet sammen. Ex: Ordet 'selskaber' skal kunne indgå i to adskilte betydninger, både som organisationer/virksomheder med de relaterede begreber aktieselskaber og anpartsselskaber og som fester med underbegreberne firmafester og jubilæer. Internt kan denne adskillelse opretholdes ved at give alle ordene i tesaurussen et id; men overfor brugeren er der behov for en anden angivelse af de forskellige betydninger. Denne angivelse kunne være overbegreberne; men det forudsætter, at alle en homografs betydninger har forskellige overbegreber. Vi har i stedet valgt noten. Når brugerne skal vælge én af en homografs betydninger, præsenteres de for de noter, der er knyttet til de forskellige betydninger, og kan så vælge ud fra dem.

Referencer. Denne relation knytter begreberne i tesaurussen til steder i lovteksterne eller noterne. Med referencerrelationen kan ordene i tesaurussen således bruges som nøgleord, der henviser til særligt relevante steder i teksterne.

Tesaurussen føjer et nyt skærmbillede til Edb-Karnov (se figur 4.12). Dette skærmbillede er nært knyttet til søge-skærmbilledet: Under formuleringen af forespørgslen kan brugeren gå ind og ud af tesaurussen for at få støtte i valget af søgeord og/eller udvide søgningen med fx relaterede begreber eller udvalgte underbegreber. Tesaurussen indeholder endvidere en facilitet til enkel nøgleordsbaseret søgning; denne facilitet giver mulighed for - uafhængigt af fuldtekstsøgefaciliteterne - at foretage opslag direkte til de steder i lovteksterne eller noterne, som referencerne henviser til.



Figur 4.12. Tesaurus-skærmbilledet. Tesaurus-skærmbilledet fokuserer på én term og viser dens relationer til de øvrige. Der kan kun knyttes én note til en term; de øvrige relationer er en-til-mange. Til højre ses funktionerne: Nyt Ord der er navigationsfunktionen, Vælg og Til Søgning der henholdsvis føjer ord til forespørgslen og skifter til søge-skærmbilledet, Se Ref der muliggør enkel nøgleordsbaseret søgning, Rediger der er redigeringsfunktionen og Til Menu der skifter til menu-skærmbilledet.

Grundtesaurus

Det kunne naturligvis helt overlades til de enkelte brugere at opbygge en tesaurus, der passede til deres behov; men det er hverken hensigtsmæssigt eller overkommeligt for brugerne. Ideen med en dynamisk tesaurus er at give brugerne mulighed for at ajourføre tesaurussen, ikke at lade dem udvikle den fra grunden. Edb-Karnov skal derfor være født med en grundtesaurus, som brugerne kan bruge uændret eller ændre, som de ønsker. Kravene til en grundtesaurus er, at den er anerkendt, at den ikke er alt for omfattende, og at den indeholder såvel juridiske begreber som 'daglig tale'. Behovet for 'daglig tale' skyldes, at der ofte er forskel på de etablerede juridiske begreber og de almindeligt anvendte. Et eksempel på dette er ‘Lov om dagpenge ved sygdom eller fødsel’, der normalt kaldes ‘sygedagpengeloven’. Et andet eksempel er ordet 'moms', der ikke forekommer i momsloven; i momsloven tales om (almindelig) omsætningsafgift og merværdiafgift.

Der findes os bekendt ikke en dansk tesaurus for det juridiske område; vi har derfor benyttet sagregistret i Virksomheds-Karnov som grundtesaurus. Sagregistret er et stikordsregister organiseret i hovedord med underbegreber i ét til tre niveauer. Registret indeholder således et hierarki på fire niveauer; det indeholder derimod ikke ret mange henvisninger til synonyme og næsten synonyme begreber og slet ingen noter. Endelig indeholder sagregistret henvisninger og giver dermed mulighed for at etablere en facilitet til nøgleords-baseret søgning. Henvisningerne er enten til en hel lovtekst - i så fald henvises til den side hvor lovteksten starter - eller til en bestemt paragraf på en side. I enkelte tilfælde henvises også til noter.

Funktionærer
- lov 780, 782
- afskedigelse **2f** 785
- - kønsdiskriminination **14f** 775
- anbefaling **17** 798
- avertering **19** 799
- bortvisning **3** 789
- død **8** 796
- foreningsforhold, pga, lov 780
- frihed t søge arbejde **16** 797
- godtgørelse v opsigelse **2a** 788, **2b** 788

Figur 4.13. Udsnit af sagregistret i Virksomheds-Karnov. Registret er organiseret i hovedord (her 'funktionærer') med underbegreber i ét til tre niveauer. Der henvises enten blot til en side eller til en bestemt paragraf på en side. Et 'f' efter paragrafangivelsen betyder paragraffen og de følgende paragraffer.

Sagregistret er ikke den ideelle grundtesaurus. Årsagen til dette er, at indgangene i sagregistret er valgt udelukkende med tanke på læseren; i Edb-Karnovs tesaurus skal de også vælges med tanke på, at de skal bruges som søgeord i fuldtekstsøgningen. Den væsentligste forskel er, at hver indgang i tesaurussen kun må bestå af ét ord. I det følgende vil vi kort beskrive den redigering af sagregistret, vi foretog for at gøre det til Edb-Karnovs tesaurus:

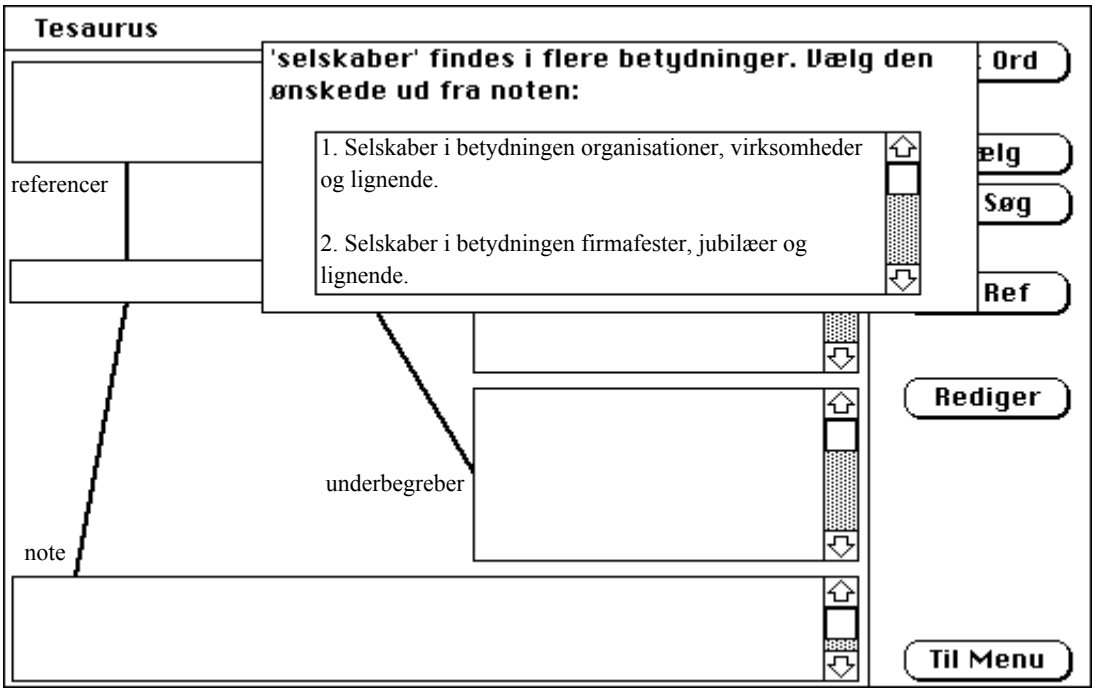
- Indgange med henvisninger til sider, der ikke er med i vores udtræk af Virksomheds-Karnov, er fjernet.
- Indgange der består af flere ord:
  - Hvis der er ét oplagt stikord, er indgangen reduceret til dette ord. Ex: 'foreningsforhold, pga, lov' (se figur 4.13) er reduceret til 'foreningsforhold'.
  - Hvis der er flere uafhængige stikord, er indgangen erstattet af en indgang for hvert af disse. Ex: 'kemikalie- og olieaffald' er erstattet med 'kemikalieaffald' og 'olieaffald'.
  - Hvis der hverken er ét oplagt eller flere uafhængige stikord, er indgangen fjernet. Ex: 'offentlig forvaltning'. Hvis indgangen havde en henvisning, er den søgt bevaret ved at føje den til et eventuelt overbegrebs henvisninger.
- Indgange af typen 'lov', 'bkg' og lignende (se figur 4.13) er fjernet; men henvisningen er føjet til overbegrebets henvisninger.
- I de tilfælde, hvor der forekommer flere forskellige bøjninger af et ord, er bøjningsformen justeret. For at støtte søgning med joker-operatorer har vi valgt den korte form (ubestemt ental). Ex: I sagregistret forekommer både 'gave' og 'gaven', her er 'gaven' erstattet med 'gave'.
- I sagregistret går henvisninger til noter ikke direkte til noten, men til den paragraf hvor notehenvisningen forekommer. Disse henvisninger er erstattet med henvisninger direkte til noten.
- I sagregistret fremgår det ikke, om '**2f**' henviser til §2 litra f eller til §2 og følgende paragraffer. Da meget få paragraffer har et litra f, vælger vi altid at opfatte '**2f**' som §2 og følgende paragraffer.

Hovedvægten i grundtesaurussen ligger på de hierarkiske relationer mellem ordene. I vores tilfælde skyldes det kilden (sagregistret i Virksomheds-Karnov); men det er faktisk meget udbredt indenfor forskningen i tesaurusfaciliteter til søgesystemer, se fx (Mili & Rada 1988) og (Strong & Drott 1986). Grundtesaurussen omfatter 466 ord.

Navigation

Tesaurus-skærmbilledet fokuserer på én term og viser dens relationer til de øvrige. Der er derfor behov for en funktion, der gør det muligt at bevæge sig rundt i tesaurussen. Navigationsfunktionen skal dels gøre det let at komme rundt, dels udformes så risikoen, for at brugeren farer vild i tesaurussen, bliver så lille som muligt.

Vi forestiller os, at der er behov for to væsensforskellige navigationsfunktioner. Den ene skal gøre det let at komme direkte til et givet begreb; den anden skal gøre det let at følge relationerne mellem begreberne. Brugeren vil ofte bruge den første til at komme ind i tesaurussen og derefter den anden til at undersøge området omkring det udvalgte begreb. Hver af disse navigationsfunktioner muliggør navigation i hele tesaurussen; men brugen af tesaurussen bliver omstændelig, hvis kun den ene af de to er til rådighed. Edb-Karnov indeholder derfor begge muligheder: Brugeren kan slå op på et begreb enten ved at taste det ind eller ved at klikke på et overbegreb, relateret begreb eller underbegreb. Den første funktion er en global udvælgelse af det begreb, der skal fokuseres på. Hvis det indtastede ord optræder i flere forskellige betydninger i tesaurussen, bliver brugeren bedt om at vælge den ønskede betydning ud fra noterne (se figur 4.14). Den anden funktion er en lokal udvælgelse af det begreb, der skal sættes i centrum. Her er homograferne aldrig skyld i tvetydigheder, da der vælges fra en liste, hvor hvert element er et begreb i én af dets muligvis flere betydninger.



Figur 4.14. Et eksempel på hvordan de forskellige betydninger af et homograf (her: 'selskaber') adskilles. Denne måde at foretage adskillelsen på er ikke speciel for navigationen; adskillelsen foregår på samme måde alle de steder i tesaurussen, hvor den er relevant. Det 'pop up'-vindue, der bruges i adskillelsen af et homografs betydninger, dækker en del af grundbilledet. Det kan næppe undgåes; men det er et problem, at Hypercard ikke giver mulighed for at definere flytbare 'pop up'-vinduer.

De to ovennævnte funktioner giver mulighed for at vælge det ord, der skal fokuseres på, ved at indtaste eller markere det. Disse to muligheder er ikke helt tilstrækkelige; det skal også være muligt at bruge tesaurussen uden på forhånd at have et ord, der kan tages udgangspunkt i. Dette problem løses ofte ved at udvide tesaurussen med en rod, der har forbindelse til alle begreberne på hierarkiets øverste niveau, se fx (McMath m.fl. 1989). Et

sådant samlende topbegreb giver adgang til hele tesaurussen ovenfra. Hvis tesaurussen er et hierarki, kan søgning i tesaurussen derefter foregå ved på hvert niveau at vælge det underbegreb, der bedst indkredser det eller de ønskede ord. Edb-Karnovs tesaurus er et netværk, ikke et hierarki; men vi mener alligevel, et samlende topbegreb vil gøre tesaurussen lettere at finde rundt i. Vi tilføjer derfor topbegrebet 'virksomhedsjura' til tesaurussen. I forbindelse med grundtesaurussen giver 'virksomhedsjura' samlet adgang til alle hovedordene i Virksomheds-Karnovs sagregister.

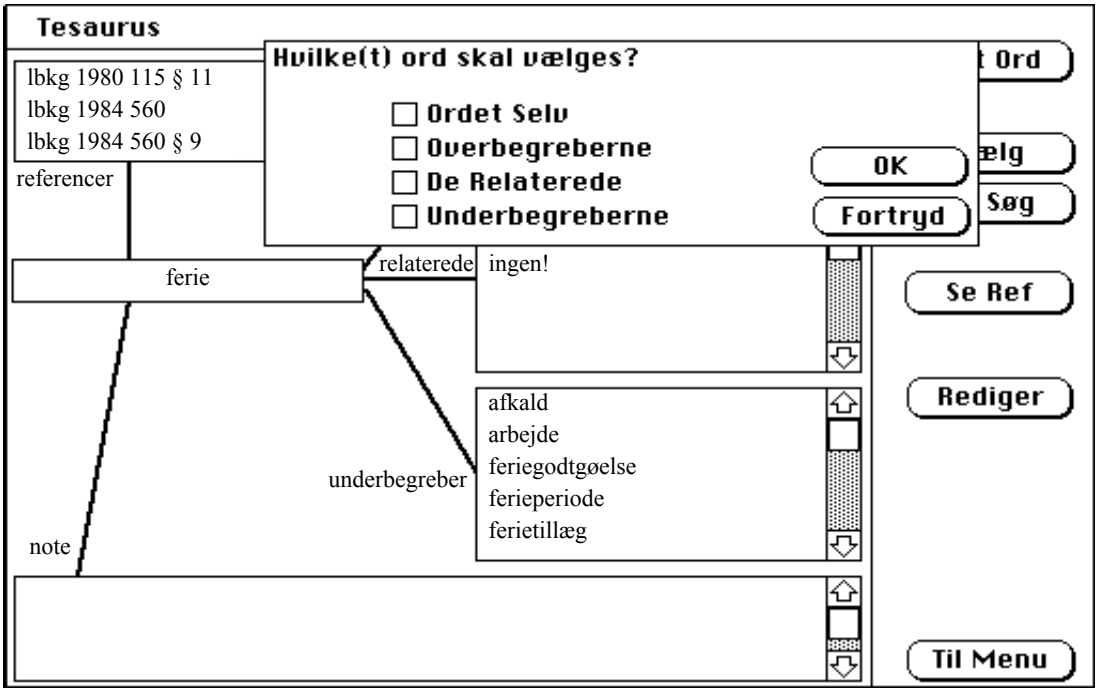
I lighed med at brugeren kan have behov for flere sagsorienterede emneregistre, kan der også være behov for flere forskellige tesaurusser. Udover grundtesaurussen kan vi forestille os, at brugeren har en tesaurus, der specielt er rettet mod hans/hendes primære sagsområde. Vi vil ikke behandle sådanne ekstra tesaurusser; men de kan umiddelbart lægges ind ved siden af grundtesaurussen. Hvis der lægges yderligere tesaurusser ind, vil det være naturligt at tilføje et fælles topbegreb for alle tesaurusserne. Underbegreberne for et sådant topbegreb, fx 'tesaurusser', vil udgøre en oversigt over de tesaurusser, systemet omfatter.

Navigationen kunne støttes ved altid at vise vejen fra det aktuelle begreb til topbegrebet. Det vil gøre det lettere for brugerne at holde orienteringen, når de bevæger sig rundt i tesaurussen. Denne facilitet er uproblematisk i tesaurusser, der er hierarkier. I Edb-Karnov vil der ofte være flere veje fra det aktuelle begreb til topbegrebet. I de tilfælde hvor brugeren har navigeret sig ned gennem tesaurussen ved at klikke på underbegreber, vil det mest oplagte være at vise den vej brugeren har fulgt fra topbegrebet til det aktuelle begreb. I de øvrige tilfælde, fx når brugeren er gået ind i tesaurussen ved at indtaste et begreb, kunne man vælge altid at angive en korteste vej til topbegrebet. Denne facilitet kan let implementeres - det største problem bliver at finde et sted på skærmbilledet, hvor der er plads til at vise den. Vi afgrænser os imidlertid fra at implementere den. Edb-Karnovs tesaurus giver således udelukkende et lokalt billede af relationerne mellem begreberne.

Faciliteter til direkte at støtte søgning

Tesaurussen omfatter to faciliteter, der er direkte knyttet til søgningerne. Den ene er rettet mod fuldtekstsøgningen, den anden mod nøgleordsbaseret søgning. Vi ser først på tesaurussens støtte af fuldtekstsøgningen.

I forbindelse med fuldtekstsøgningen kan tesaurussen bruges til at støtte valget af søgeord og/eller supplere de valgte søgeord med fx relaterede begreber. Begge dele sker ved, at udvalgte ord fra tesaurussen føjes til forespørgslen. De to forskellige måder at bruge tesaurussen på - udvælgelse og supplering - har givet anledning til to forskellige funktioner. Det er for det første muligt at klikke på et enkelt ord og derefter føje det til forespørgslen ved at klikke på Vælg. For det andet er det muligt at tilføje hele gruppen af fx underbegreber; denne mulighed kommer frem, når brugeren aktiverer Vælg uden først at have markeret et ord (se figur 4.15). Disse to funktioner er ikke de eneste, det kunne være relevant at have. Den mest oplagte funktion at tilføje er en mulighed for globalt at kunne slå tilføjelse af fx relaterede begreber til og fra. Vi har undladt denne funktion; men på baggrund af de to funktioner, vi har med, er den let at tilføje.



Figur 4.15. Hvis Vælg aktiveres uden, at der først er markeret et ord, fremkommer denne menu. Et eller flere af menupunkterne kan vælges, før menuen forlades, eller hele funktionen kan fortrydes.

I LEXIS er det (med kommandoen 'explode') muligt at udvide søgningen med ikke blot de direkte underbegreber, men også underbegrebernes underbegreber osv (Harrison 1981). Efter vores mening udvider denne facilitet søgningen i et sådant omfang, at det bliver svært for brugeren at overskue, hvad forespørgslen egentlig dækker. Vi vurderer, at denne facilitet går langt ud over, hvad brugerne normalt er interesserede i. I Edb-Karnov betyder tilføjelse af underbegreber derfor kun tilføjelse af de direkte underbegreber. Denne funktion kan enten implementeres ved at tilføje hver enkelt underbegreb til forespørgslen for sig eller ved at tilføje noget i retning af 'underbegreber(miljøbeskyttelse)' og overlade resten til selve søgningen. Den første mulighed har den fordel, at brugeren kan redigere i de valgte ord; den anden mulighed har den fordel, at fx et begrebs underbegreber kan føjes til forespørgslen, uden at brugeren behøver gå ind i tesaurussen. Vi har valgt den første mulighed, fordi den er lettest at implementere. Hovedproblemet ved den anden mulighed er, at ikke blot fremfindingen af underbegreberne, men også håndteringen af homografer, flyttes over i selve søgningen.

Tesaurussen indeholder også en facilitet til en enkel form for nøgleordsbaseret søgning. Nøgleordsbaseret søgning kan enten ske ved at hæfte nøgleord på dokumenter eller ved at hæfte dokumenter på nøgleord. Den første type er udbredt, se fx (Belkin & Croft 1987) og (Salton & Buckley 1988). Her erstattes søgning i selve dokumentets tekst af søgning i den mængde nøgleord, der er hæftet på det. Den anden type, der knytter dokumenter til nøgleord, giver mulighed for opslag. Her kan brugeren direkte - dvs uden forudgående søgning - se, hvilke dokumenter, der har et givet begreb som et hovedemne. Vi finder, at den sidste type nøgleordsbaseret søgning bedst kompletterer fuldtekstsøgning, og implementerer derfor den i Edb-Karnov.

I Edb-Karnov giver faciliteten til nøgleordsbaseret søgning mulighed for opslag på de dokumenter, der er indekseret med det pågældende ord. Det er ikke muligt at kombinere nøgleordene til forespørgsler. En forudsætning, for at denne type nøgleordsbaseret søgning kan fungere, er, at søgesystemet indeholder faciliteter, der kan hjælpe brugeren med at bevare overblikket over de eksisterende nøgleord. Vi mener, tesaurussen er

velegnet til dette formål, dels fordi en stor del af ordene i tesaurussen er velegnede nøgleord, dels fordi tesaurussen udgør et netværk, der anskueliggør relationerne mellem ordene.

Vi inddrager nøgleordsbaseret søgning både i erkendelse af og som understregning af, at ethvert godt søgesystem må indeholde en kombination af søgeteknikker. Vi nøjes med at etablere muligheden for såvel fuldtekstsøgning som nøgleordsbaseret søgning. Der ligger imidlertid en stor udfordring i at integrere fuldtekstsøgning og nøgleordsbaseret søgning på en frugtbar måde. Det er et område, der er ved at få en central placering i forskningen om søgesystemer (Belkin & Croft 1987). En sådan integration skal være i stand til at støtte både en *bottom-up* og en *top-down* søgestrategi.

Nøgleordene kan ikke blot hæftes på hele tekster, men også på teksternes grundenhed. Referencerne kan således enten henvise til en lovtekst i sin helhed eller til en specifik paragraf eller en specifik note. Nøgleordsbaseret søgning aktiveres ved at klikke på en reference og derefter på Se Ref. Resultatet af dette er et opslag på den pågældende tekst.

Et af de store problemer ved nøgleordsbaseret søgning er normalt, at såvel mængden af nøgleord som de nøgleord, der er hæftet på en given tekst, er prædefinerede og ikke kan ændres af den enkelte bruger. I Edb-Karnov kan brugeren til enhver tid føje et nyt ord til tesaurussen eller oprette en ny reference fra et af tesaurussens begreber.

Redigering

Det er med redigeringsfunktionen, tesaurussen bliver dynamisk. Samspillet mellem søgning og redigering er imidlertid afgørende for, hvordan dynamikken opleves, og hvor meget den anvendes. Redigeringen kan etableres som en seperat aktivitet adskilt fra søgningerne. Det forudsætter, at brugeren fra tid til anden lægger sine andre opgaver til side og tager sig tid til at ajourføre tesaurussen. Dorthe la Cour giver udtryk for, at hun vil foretrække noget i denne retning. Vi frygter, det vil betyde, at dynamikken ikke bliver brugt. Hvis redigeringen ikke skal være en seperat aktivitet, kan den i stedet integreres med søgningerne. Ideen er her, at ajourføringerne foretages løbende og udføres, når brugeren alligevel er inde i tesaurussen i forbindelse med en søgning. Vi vælger at integrere redigering og søgning. De brugere, der ønsker det, kan stadig gå ind i tesaurussen udelukkende for at redigere eller udelukkende for at søge; men muligheden for at kombinere de to er også til stede.

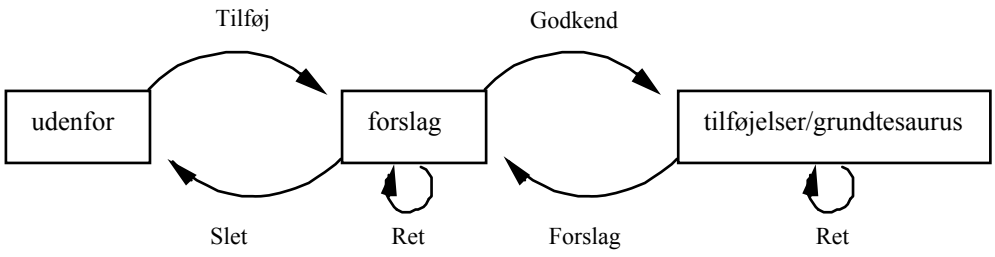
Hvis dynamikken skal kunne udnyttes, uden at tesaurussen hurtigt vokser sig stor, rodet og uoverskuelig, så skal brugeren kunne føre en vis kontrol og justits med, hvilke ajourføringer der foretages. Det kan gøres med en logfil, hvor alle ajourføringer automatisk noteres. Vi lægger imidlertid vægt på, at det foregår interaktivt og indenfor integrationen af redigering og søgning, og har derfor valgt en anden løsning. Det, der er behov for, er efter vores mening, at ajourføringerne kan fungere i en prøvetid og derefter tages op til endelig vurdering. Vi har implementeret dette ved at opdele ordene i tesaurussen i to grupper. Den første gruppe omfatter de ord, der stammer fra grundtesaurussen eller er blevet tilføjet senere; den anden gruppe indeholder forslag. Den enkelte jurist kan have glæde af denne opdeling; men den ville være endnu mere relevant, hvis Edb-Karnov var et flerbruger-system. I så fald kunne forslagene kommunikeres mellem de forskellige brugere og godkendes/forkastes ud fra en fælles vurdering.

Ideen med forslagene er, at begreber, der føjes til tesaurussen, i første omgang får status af forslag. Tilsvarende slettes begreber ikke umiddelbart fra tesaurussen; de reduceres i første omgang til forslag. På skærmen er forslag markeret ved, at de pågældende ord står i parentes. Det er således let at skille de nyligt foretagne ajourføringer fra tesaurussens etablerede begreber. Ajourføringerne kan blive stående som forslag, så længe det skal være; men det er tanken, at brugeren efter en tid enten godkender eller

sletter dem. Denne beslutning kunne støttes ved, at systemet holdt tal på, hvor mange gange et forslag blev benyttet. Det ville give en vis indikation af, om forslaget var velvalgt eller overflødigt. Vi afgrænser os fra denne statistikfacilitet, da der ligger en del arbejde i at implementere den. Ex: Når brugeren indtaster et ord, som findes i tesaurussen i flere forskellige betydninger, kan systemet ikke afgøre, hvilken en af betydningerne der benyttes.

Forslag bruges ikke blot i forbindelse med tilføjelse og sletning af ord; de bruges også ved tilføjelse og sletning af referencerne og relationerne mellem ordene. Når fx et overbegreb står i parentes er det således relationen, ikke selve ordet, der er et forslag. Et forslag kan markeres og tilføjes til forespørgslen på lige fod med de godkendte begreber. Når en gruppe begreber, fx et givet begrebs underbegreber, tilføjes til forespørgslen, er situationen lidt anderledes. Brugeren kontrollerer - formodentlig - ikke, hvilke ord det præcis er, der tilføjes. Det er den overordnede, intuitivt forståelige funktionalitet 'med underbegreber', han/hun går efter. Vi har derfor valgt at gøre det sådan, at det kun er de godkendte underbegreber, der tilføjes til forespørgslen. Forslagene holdes udenfor, netop fordi de kun er forslag.

Redigeringsfunktionen omfatter fem underfunktioner: Tilføj, Ret, Slet, Godkend og Forslag. Ret er den eneste af de fem, der ikke allerede er beskrevet. Ret giver mulighed for at foretage en global ændring af et ords stavemåde eller ændre en notes indhold med udgangspunkt i den eksisterende note. Den nedenstående figur illustrerer de fem redigeringsfunktioners funktionalitet:



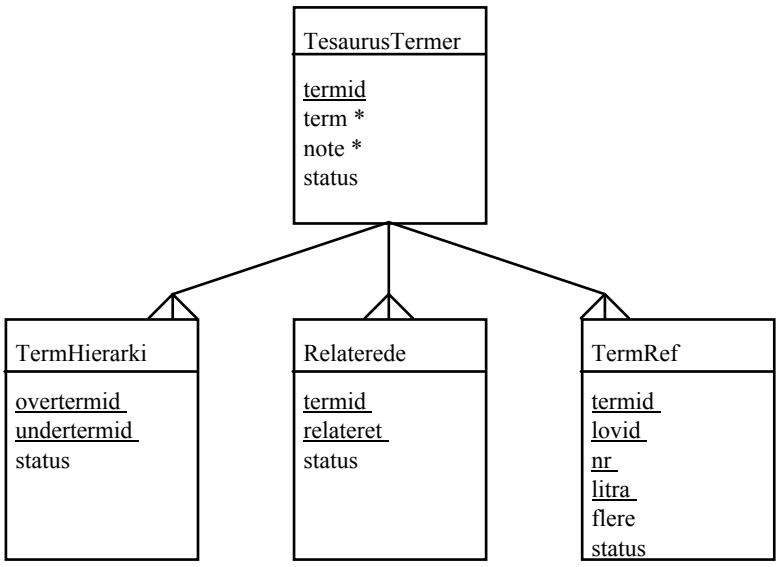
Figur 4.16. Redigeringsfunktionerne. I forhold til tesaurussen kan et begreb være udenfor, være lagt ind som forslag eller være en godkendt tilføjelse eller del af grundtesaurussen. Figuren illustrerer funktionaliteten i de fem redigeringsfunktioner. Med Tilføj kan nye begreber føjes til tesaurussen som forslag; med Godkend kan forslag gøres til etablerede tilføjelser; med Forslag kan etablerede begreber reduceres til forslag; med slet kan forslag slettes; og med Ret kan stavemåden for et vilkårligt ord i tesaurussen ændres.

Redigeringsfunktionen aktiveres ved at klikke på det ord, der skal redigeres, (ved Tilføj er det nok at klikke i det felt, fx overbegreber, hvor tilføjelsen skal ske) og derefter klikke på Rediger. Herved præsenteres brugeren for en menu med redigeringsfunktionerne. Det er muligt at vælge flere af disse funktioner efter tur; flere begreber kan således tilføjes umiddelbart efter hinanden, eller et begreb kan reduceres til forslag og straks slettes.

Datamodel

Vi sammenfatter dette afsnit med en kort beskrivelse af de fire tabeller, tesaurussen giver anledning til (se figur 4.17). I datamodellen kan adskillelsen af forslagene og de godkendte begreber ske enten ved at have to sæt tabeller eller ved at have ét sæt tabeller med en statusattribut. Forslagene og de godkendte begreber bruges næsten altid sammen - når forslagene behandles, behandles de godkendte begreber også, og omvendt. Af den grund vælger vi en statusattribut; det giver adgang til både forslagene og de godkendte begreber i samme SQL-sætning.

Internt i tesaurussen identificeres ordene ved et entydigt nummer - et termid. Det muliggør såvel eksistensen af homografer som globale rettelser af ordenes stavemåde. Sammenknytningen af termer og termid'er sker i tesaurussens grundtabel - *tesaurustermer*. Denne tabel indeholder alle tesaurussens begreber samt de noter, der eventuelt er knyttet til dem. Over- og underbegreber giver anledning til tabellen *termhierarki*. Denne tabel sammenknytter par af termid'er, sådan at det første begreb er overbegreb for det andet. Det samme begreb vil typisk forekomme som overbegreb i nogle tupler og som underbegreb i andre. De relaterede begreber giver anledning til en tabel, der i opbygning svarer fuldstændig til den foregående: *Relaterede*. Endelig giver referencerne anledning til tabellen *termref*. Da referencerne ikke blot kan være til hele lovtekster, men også til specifikke paragraffer eller noter, kræver det fire attributter at udpege den tekst, der henvises til: type, lovid, nr og litra. Type-attributten angiver, om der er tale om en lovtekst eller en note, og afgør, om nr og litra angiver paragrafnr og paragraflitra eller notenr og notelitra. Tabellen indeholder også en flere-attribut, der angiver om henvisningen kun er til den udpegede tekst eller også omfatter de følgende paragraffer/noter. De fire tabeller er alle på såvel tredje som fjerde normalform.



Figur 4.17. Datamodel for Edb-Karnovs tesaurus. Internt identificeres tesaurussens termer ved et entydigt nummer - et termid. Tabellernes primærnøgle er markeret ved understregning, alternative nøgler med stjerner.

4.7 Egne notater

De to foregående afsnit har behandlet faciliteter til søgning i de tekster, der ligger i Edb-Karnov. Vi ser imidlertid også et stort behov for, at brugeren løbende kan tilføje nye tekster i form af egne notater. Disse notater skal ikke udgøre en isoleret del af Edb-Karnovs tekster, men integreres med de øvrige. Det er en enkel, oplagt og letforståelig idé, og da Edb-Karnov er baseret på en relationsdatabase, er tilføjelse af nye tekster også enkelt set ud fra et teknisk synspunkt. Af Edb-Karnovs faciliteter til at støtte udvikling over tid er det denne, Dorthe la Cour og Per Sjøqvist umiddelbart reagerer mest positivt på.

Notater minder på mange måder om noter; mange af de funktioner, der er relevante i forbindelse med noter, er således også relevante i forbindelse med notater. Vi fokuserer i det følgende på det, der er specielt for notater. Afsnittet indledes med en diskussion af

ideen med at give mulighed for tilføjelse af egne notater, så behandles oprettelse og redigering af notaterne og derefter notathenvisningernes placering i teksterne. Efter det diskuteres muligheden for at bruge notater som bogmærker. Endelig afsluttes afsnittet med en kort beskrivelse af de udvidelser af datamodellen, notaterne giver anledning til.

Ideen

Edb-Karnovs lovtekster og noter er ikke de eneste tekster, det er relevant at søge juridisk information i. En årsag til dette er, at brugerne ofte har behov for endnu mere detaljerede oplysninger om retskildernes indhold end dem, de kan finde i Edb-Karnov. Brugerne søger derfor også i andre kilder, fx kilder rettet specielt mod deres område indenfor virksomhedsjuraen. En anden årsag er, at brugerne ikke kun søger information i retskilderne, men fx også i deres egne notater fra tidligere, lignende sager. Et juridisk informationssøgesystem bør derfor løbende kunne udvides med tekster, som brugeren finder relevante i forbindelse med sit arbejde. Indholdet af disse tekster er helt op til brugeren; det kan fx være tidligere udarbejdede kontrakter, referater af cirkulærer eller blot henvisninger til relevante kilder.

Der ligger to ideer bag muligheden for at føje egne notater til Edb-Karnov: For det første skal notaterne være søgbare på fuldstændig samme måde som lovtekster og noter. Baggrunden for dette er, at notaterne skal sidestilles med de øvrige tekster - notaterne adskiller sig kun fra de øvrige tekster ved at være skrevet af brugeren. Søgefaciliteterne skal således også kunne bruges til fremfinding af notater. Det indbefatter, at søgningerne skal kunne afgrænses til kun at omfatte notater, ligesom der skal kunne søges i notater i samme arbejdsgang, som der søges i de øvrige tekster.

For det andet giver notater mulighed for, at alle slags tekster kan systematiseres på en ny måde. Notater, kontrakter og lignende systematiseres ofte kronologisk eller efter journalnummer; med Edb-Karnov er der mulighed for at bruge strukturen i lovgivningssystemet som systematik. Notaterne adskilles ikke fra de øvrige tekster, men knyttes - ligesom noterne - altid til en lovtekst. Notaterne kan naturligvis være knyttet direkte til lovtekstens indhold; men muligheden for at tilføje nye tekster betyder, at Edb-Karnovs potentielle anvendelsesområder udvides væsentligt. Notaterne kan fx bruges til at etablere referencer fra Edb-Karnov til et journaliseringssystem og giver dermed en tilgang til journaliseringssystemets sager, som følger strukturen i lovgivningssystemet.

Med egne notater bliver det i langt højere grad muligt at integrere Edb-Karnov i juristens arbejdssituation. Et aspekt af dette er integrationen af søgesystemet og de andre systemer til kontorautomation, primært tekstbehandling. Det er et aspekt, der efterhånden ofres megen opmærksomhed, se fx (Firnberg 1986) og (Smeaton & van Rijsbergen 1986). Et andet aspekt af integrationen er muligheden for både at bruge Edb-Karnov som søgesystem og som arkiveringssystem. Det vil betyde, at væsentlige dele af juristens informationssøgning samles på ét sted. Som eksempel er grundlaget for udarbejdelse af nye kontrakter først og fremmest tidligere udarbejdede kontrakter samt organisationens og juristens notater omkring indgåelse af kontrakter. Denne type juridisk sagsbehandling - vi kalder den fremadrettet sagsbehandling - involverer derimod ikke ret meget søgning i lovteksterne. Muligheden for at føje egne notater til Edb-Karnov betyder, at tidligere udarbejdede kontrakter, notater og lignende kan samles i Edb-Karnov.

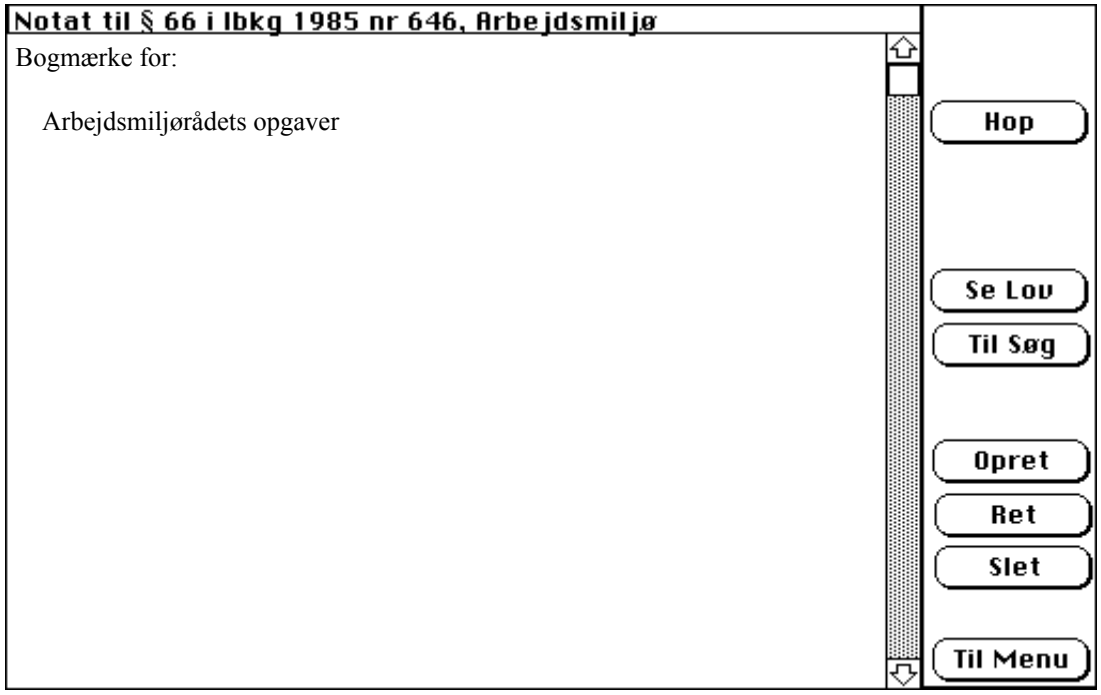
Vi gør os ingen forestillinger, om at alle relevante oplysninger på denne måde vil blive samlet i Edb-Karnov. Det er efter vores mening udelukket af såvel praktiske som principielle grunde. Til de praktiske grunde hører, at der til stadighed vil stå mange vigtige ting i margenen på gode bøger, på løse ark i ringbind osv. Det er ikke overkommeligt - og for dens sags skyld heller ikke ønskeligt - at lave om på dette forhold. Derudover er den viden, eksperterne baserer deres handlinger på, principielt umulig at formalisere og sætte

ord på (Winograd & Flores 1986), (Dreyfus & Dreyfus 1986). Store dele af den viden, jurister baserer deres arbejde på, er således ikke engang skrevet ned; den eksisterer kun i form af den enkelte jurists erfaring, intuition og ekspertise. Da Edb-Karnov ikke indeholder alle relevante oplysninger, er der behov for nogle retningslinier om, hvilke dokumenter og oplysninger, systemet indeholder, og hvilke der skal søges andre steder.

Oprettelse og redigering af notater

I Edb-Karnov kan der kun oprettes notater i tilknytning til lovteksterne. Det skyldes, at en af hovedideerne med notaterne er, at de skal give mulighed for at systematisere alle slags tekster ud fra strukturen i lovgivningssystemet. Notaterne kan imidlertid også bruges til at knytte kommentarer og bemærkninger direkte til indholdet af de enkelte tekster i Edb-Karnov. I denne sammenhæng ville det være relevant også at kunne oprette notater i tilknytning til noterne. Det er en oplagt udvidelse; men vi undlader den, da den ikke indeholder noget egentligt nyt - notater til noterne svarer fuldstændig til notater til lovteksterne.

Notaterne giver anledning til udvidelser af flere af de eksisterende skærmbilleder: Søge-skærmbilledet udvides, så søgninger også kan omfatte notater; tesaurus-skærmbilledet udvides, så referencerelationen også kan henvise til notater; og lovtekstskærmbilledet udvides, så der kan slåes op på notater og notathenvisningerne kan indplaceres i lovteksten. Derudover giver notaterne anledning til ét nyt skærmbillede (se figur 4.18). Dette skærmbillede bruges til at vise notater i deres fulde tekst, og det er her redigeringen af notaterne foretages.



Figur 4.18. Notat-skærmbilledet (her et bogmærke-notat til §66 i lbkg 1985 nr 646 om arbejdsmiljø). Funktionerne til højre giver mulighed for at hoppe til et andet af lovtekstens notater, at slå den henvisende paragraf op, at skifte til søge-skærmbilledet, at oprette, rette i og slette notater samt at skifte til menu-skærmbilledet. Det ville også være relevant med bladringsfunktioner svarende til dem på note-skærmbilledet; det har vi imidlertid afgrænset os fra.

Flere af notat-skærmbilledets funktioner svarer til dem på note-skærmbilledet; i det følgende behandles redigeringsmulighederne. For brugeren omfatter oprettelse af et notat

to aktiviteter: Brugeren skal dels angive notathenvisningens placering, dels selve den tekst der udgør notatet. Det mest hensigtsmæssige er at lade det være op til brugeren at afgøre, hvilken rækkefølge de to aktiviteter skal udføres i. Vi afgrænser os imidlertid til at implementere den ene af de to muligheder. I Edb-Karnov vælger brugeren først den paragraf, notatet skal knyttes til, ved fra lovtekst-skærmbilledet at klikke på funktionen Opret Notat. Derved skifter systemet til notat-skærmbilledet, hvor brugeren kan indtaste notatet. Oprettelsen af notatet afsluttes ved at klikke på funktionen Opret.

En stor del af notaterne må formodes at være tekster, der er udfærdiget udenfor Edb-Karnov, fx tidligere udarbejdede kontrakter. Værdien af notaterne er derfor afhængig af, at det er let at lægge disse tekster ind i Edb-Karnov, specielt at de ikke skal tastes ind igen. Macintosh'erne har en generel facilitet, klippebordet, til kopiering af tekster. Klippebordet giver såvel mulighed for kopiering af tekst indenfor en applikation som på tværs af applikationerne. Klippebordet fungerer også i forbindelse med Edb-Karnov. Det er således muligt at kopiere en tekst fra fx et tekstbehandlingssystem ud på klippebordet, derefter gå ind i Edb-Karnov og kopiere teksten fra klippebordet ind i et notat. Det er ligeledes muligt at kopiere notater såvel som paragraffer og noter den anden vej.

Det er også muligt at slette notater. Det sker ved at finde notatet frem og klikke på funktionen Slet. Endelig er det muligt at rette i eksisterende notater. Det foregår ved at finde notatet frem og derefter foretage rettelserne direkte i den viste notattekst. For at give brugeren mulighed for at fortryde påbegyndte rettelser skal det eksplicit angives, at de udførte rettelser faktisk ønskes gennemført. Det sker ved at klikke på funktionen Ret. Derved slettes det gamle notat, og det rettede lagres og inverteres.

Det er ikke kun notaternes tekst, der skal kunne rettes i; et notat kan også være placeret et forkert sted. Der er således behov for at kunne flytte notathenvisninger fra en paragraf til en anden. I Edb-Karnov kan det kun ske ved at kopiere notatteksten ud på klippebordet, slette notatet, finde den paragraf frem som skal henvise til notatet og oprette et nyt notat, hvor teksten kopieres ind fra klippebordet. Hvis flytning af notathenvisninger forekommer ofte, kan en funktion specielt til det formål let implementeres.

Notathenvisninger

Når et notat oprettes, skal der placeres en henvisning til det i lovteksten. Den måde, der henvises til noter på, er efter vores mening udmærket; vi giver derfor notathenvisningerne en tilsvarende form. I lovteksten markeres en notathenvisning således ved: '[nr]'. Hakparenteser forekommer ikke i lovteksterne iøvrigt. For at øge overskueligheden kan notaternes numre holdes fortløbende ned igennem hver enkelt lovtekst; det afgørende er imidlertid, at numrene holdes forskellige. Vi afgrænser os til at holde notatnumrene forskellige.

For noternes vedkommende kan henvisningerne være placeret hvor som helst i lovteksternes paragraffer; det giver mulighed for at knytte noter til så lille en enhed som det enkelte ord. Den samme frihed er at foretrække ved placering af notathenvisninger; den er imidlertid besværlig at implementere. Besværlighederne skyldes, at grundenheden for lagringen af lovteksterne - paragrafferne - er større end den mindste enhed, notaterne skal kunne hæftes på. Dette problem løses ikke ved et andet valg af grundenhed for lagringen; det ville kræve at grundenheden blev det enkelte ord, hvilket er helt urealistisk.

En mulighed for at løse problemet er at skrive notathenvisningen ind i paragraffens tekst. Det svarer til den måde, notehenvisningerne fungerer på: Den henvisende paragraf kan slåes op i databasen; men henvisningens placering i paragraffen fremgår kun af paragraffens tekst. Denne løsning er imidlertid uhensigtsmæssig, da notater i modsætning til noter kan tilføjes og slettes løbende. En anden mulighed er at udvide den angivelse af notathenvisningens placering, der findes i databasen, så den ikke blot udpeger den

henvisende paragraf. Det kan gøres ved at udvide angivelsen med en oplysning om, hvor mange tegn eller ord inde i paragraffen, henvisningen forekommer. Vi mener, dét er den rigtige løsning. I Edb-Karnov afgrænser vi os imidlertid til noget væsentligt simplere: Et notat knyttes altid til en paragraf, og henvisningen placeres efter paragraffens tekst. Angivelsen af henvisningens placering skal således kun omfatte paragraffen. Det er stadig muligt at hæfte flere notater på samme paragraf, i så fald placeres alle henvisninger samlet efter paragraffens tekst.

Bogmærker

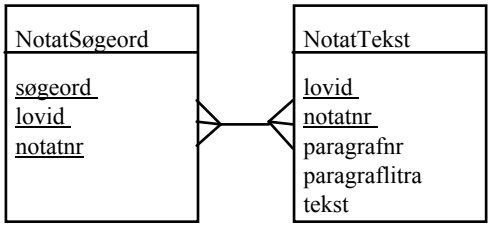
Muligheden for at komme fra notatet til den henvisende paragraf betyder, at notater kan bruges som bogmærker. Hvis brugeren må afbryde arbejdet eller af en anden grund gerne vil kunne vende tilbage til et bestemt sted i lovteksten, kan han/hun hæfte et bogmærke-notat på den aktuelle paragraf. Bogmærke-notater er helt almindelige notater; det er kun den måde, de bruges på, der adskiller dem fra andre notater. Et bogmærke-notat kan fx se ud som vist i figur 4.18. Hvis alle bogmærke-notater opbygges ud fra en skabelon i stil med den i figuren, så kan et opslag på bogmærket om emnet X foretages ved en søgning i notaterne med forespørgslen 'bogmærke og X'. Endvidere vil en søgning med det ene søgeord 'bogmærke' give en oversigt over alle bogmærker.

Bogmærker ligner på flere måder tesaurussens reference-relation - den relation der udgør Edb-Karnovs facilitet til nøgleordsbaseret søgning. Bogmærker giver imidlertid fuld frihed i ordvalget og længden af beskrivelsen af emnet. Vi forestiller os, at bogmærkerne vil kunne yde et væsentligt bidrag til ajourføringen af tesaurussens referencerelation. Bogmærker, der over en periode har vist sig relevante, kan efterhånden overføres til reference-relationen. Hermed antyder vi en 'arbejdsdeling': Bogmærkerne er af en midlertidig natur, mens tesaurussens referencer er mere permanente. Denne deling skal ikke gennemføres konsekvent; men vi tror med fordel, den kan bruges som rettesnor.

Bogmærke-notater kan også bruges til at markere alle de steder, der er relevante i forbindelse med en given sag. På denne måde kan bogmærker bruges til at fastholde og dokumentere, hvad der er grundlaget for sagen. Denne brug af bogmærker kan støttes yderligere ved automatisk at registrere de lovtekster, der findes frem i forbindelse med en given sag. Denne facilitet findes ikke i Edb-Karnov i øjeblikket. Det skyldes, at egne notater er en helt generel mulighed for at tilføje tekster og derfor ikke er rettet mod nogen bestemt anvendelse. Vi har afgrænset os til denne generelle mulighed; men bogmærker er et eksempel på, at der også er behov for specielle typer notater med tilhørende funktioner.

Datamodel

Notaterne giver anledning til de to sidste tabeller, vi vil føje til Edb-Karnovs datamodel (se figur 4.19). Hvert notat identificeres entydigt ved et lovid og et notatnr. Den første tabel, *notattekst*, indeholder den tekst, der udgør notatet, samt en identifikation af den henvisende paragraf. Denne tabel bruges, når et notat eller en paragraf skal præsenteres for brugeren i sin fulde tekst; ved præsentation af paragraffer er det eventuelle notathenvisninger, der hentes i tabellen. Den anden tabel, *notatsøgeord*, indeholder en tupel for hvert ikke-stopord, der forekommer i notaterne, samt en angivelse af, hvilket notat ordet forekommer i. Denne tabel bruges i forbindelse med søgningerne. Disse to tabeller er på tredje såvel som fjerde normalform. Det betyder, at hele Edb-Karnovs database er på tredje og fjerde normalform.



Figur 4.19. De to tabeller som tilføjelse af notater giver anledning til. Disse to tabeller er helt analoge til de to, noterne gav anledning til. Indenfor en lovtekst identificeres et notat entydigt ved et nummer - her er intet behov for at tilføje et litra. Tabellernes primærnøgle er markeret ved understregning.

4.8 Afprøvning og optimering

Vores prototype Edb-Karnov er færdigudviklet. Vi har hidtil mest koncentreret os, om udformningen af de faciliteter, der skal støtte Edb-Karnovs udvikling over tid. Edb-Karnov har imidlertid også til formål at give os mulighed for at vurdere, om det er realistisk at basere fuldtekstsøgesystemer med en funktionalitet som Edb-Karnovs på relationsdatabaser. I dette afsnit vil vi afprøve prototypens svartider og se på dens pladsforbrug.

Afsnittet starter med en diskussion af målet med afprøvningen. Derefter beskriver vi de afprøvningstilfælde, der er grundlaget for afprøvningen. Efter det diskuteres de resultater, afprøvningen har givet, og de optimeringer, vi har foretaget for at opnå dem. Afsnittet afsluttes med en konklusion på afprøvningen.

Målet med afprøvningen

Målet med afprøvningen er at nå frem til en vurdering af, om det er realistisk at basere fuldtekstsøgesystemer på relationsdatabaser. Relationsdatabaser kritiseres for at bruge for meget tid og plads til at være velegnede som grundlag for søgesystemer. Vi lægger i denne afprøvning hovedvægten på svartiderne, mens pladsforbruget kun behandles kort.

Det er almindeligt, at kravet til svartiderne er ens for enkle og omfattende søgninger (fx to sekunder). Det svarer efter vores mening ikke til brugernes oplevelse af svartiderne. De enkle søgninger skal naturligvis give anledning til korte svartider. Ved de omfattende søgninger er svartiderne, som vi var inde på i afsnit 4.1, ikke nær så kritiske. Brugerne ved, hvornår de sætter omfattende søgninger i gang, og er derfor forberedt på, at det kan tage lidt tid. Det er den væsentligste grund, til at svartiderne for de komplicerede søgninger ikke er så kritiske: De lidt længere svartider kommer ikke bag på brugerne. Derudover spiller det også ind, at svaret har værdi for brugerne, og at det som regel vil tage meget, meget længere tid at foretage søgningen manuelt.

Med Edb-Karnov har det været målet at udvikle en prototype hvor svartider er rimelige til demonstrationsbrug. Vi har - lidt tilfældigt - udmøntet dette i et krav om, at svartiderne for de omfattende søgninger ikke må overstige 1 minut. Vi har således ikke fokuseret på opnåelse af optimale svartider. Baggrunden for det er, dels at formålet med specialet er et andet, dels at det kræver indgående kendskab til selve Oracle-databasesystemet, og et sådant kendskab har vi ikke.

Vi har opdelt de søgninger, der udgør afprøvningen, i enkle, medium og omfattende. Vi tror, det er enkle og medium søgninger, brugeren oftest vil foretage. Afprøvningen skal helst vise, at enkle søgninger giver korte svartider, at medium søgninger kun giver lidt længere svartider, og at svartiderne for de omfattende søgninger ikke overstiger 1 minut.

Afprøvningstilfælde

Vi vil ikke her beskrive hvert afprøvningstilfælde, men se på de faktorer, der afgør om en

søgning er enkel, medium eller omfattende. Den mængde tekst, der skal gennemsøges, har naturligvis betydning for svartiden. Det er hurtigere at søge efter et ord i én lovtekst end i 100. Vi formoder, at det antal gange et søgeord forekommer i tekstmængden også har betydning for svartiderne. Det er hurtigere at finde de tekster, der matcher forespørgslen, hvis søgeordet forekommer én gang fremfor 50. Derudover afhænger svartiderne af antallet af operatorer i forespørgslen. Det tager længere tid at afgøre, om en tekst matcher forespørgslen, hvis der er mange operatorer og dermed mange søgeord, end hvis der er få. Endelig må det også påvirke svartiden, om nærhedsoperatoren for OG-operatoren er 'indenfor samme lovtekst' eller 'indenfor samme paragraf'. Vi har altså fire parametre, der kan varieres i opstillingen af afprøvningstilfældene: Tekstmængden, søgeordenes frekvens i teksterne, antallet af boolske operatorer og nærhedsoperatoren. De 30 afprøvningstilfælde, som opstod ved variation af de fire parametre, er gengivet ogbeskrevet i bilag 5.

De 6 første afprøvningstilfælde i bilag 5 vil vi karakterisere som enkle søgninger, da der udelukkende er tale om søgning i én lovtekst. De næste 8 tilfælde er medium søgninger, der omfatter alle lovtekster og noter i et mindre retsområde. De sidste 16 tilfælde er omfattende søgninger, da der enten søges i et stort retsområde eller i alle lovtekster. En grov inddeling af søgningerne i enkel, medium og omfattende kan i første omgang baseres på tekstmængden. En anden væsentlig parameter er imidlertid antallet af operatorer. De mest omfattende søgninger bliver dem, hvor både tekstmængden er stor, og antallet af boolske operatorer er stort (42 boolske operatorer er det højeste, vi prøver med).

Vi har udført flere forsøg med de 30 afprøvningstilfælde. De 400 sider fra Virksomheds-Karnov er lagt ind ca 100 sider ad gangen. Når hundrede sider var lagt ind, blev de 30 afprøvningstilfælde udført på denne tekstmængde, derefter blev yderligere hundrede sider lagt ind. Resultaterne udgør de fire første forsøg i bilag 5. Det er gjort for at se, hvordan svartiderne udvikler sig med tekstmængden. I Oracle-manualerne anbefales det, at alle data i en tabel lægges ind på én gang; det skulle give de bedste svartider. For at se, hvor stor betydning det har, fulgte vi anbefalingen; resultatet er forsøg 5 i bilag 5.

Resultatet af afprøvningen

De svartider, afprøvningen har givet, er fundet ved at aflæse det interne ur i Hypercard umiddelbart før og efter udførelsen af SQL-forespørgslen. Dette ur kan kun aflæses i sekunder. En forespørgsel, der skal søge i både lovtekst, noter og notater, består af tre SQL-forespørgsler. I disse tilfælde er svartiden fundet ved at summere svartiderne for de tre SQL-forespørgsler; det giver anledning til afrundingsfejl. Forskelle i svartiderne på et par sekunder kan således lige så godt skyldes afrundingsfejl i tidtagningen som reelle forskelle i svartiden. I det følgende vil vi se på de resultater, afprøvningen gav.

De fire første forsøg (se bilag 5) skulle vise, hvad indlæggelse af større og større tekstmængde betød for svartiderne. Hvis der ikke var indeks på søgeordstabellerne, måtte sammenhængen mellem den indlagte tekstmængde og svartiderne forventes at være lineær. Med indeks må sammenhængen forventes at være logaritmisk. Når indeksene bliver meget store, kan Oracle muligvis få svært ved at håndtere dem, eller det kan fx blive nødvendigt at flytte et indeks fra RAM til disk. Det vil vise sig som et knæk på kurven over sammenhængen mellem tekstmængde og svartid. Et sådant knæk betyder, at svartiden pludselig bliver væsentligt højere end ved fortsættelse af den logaritmiske sammenhæng.

Stigningen i tekstmængden fra 100 til 400 sider har typisk givet en stigning i svartiderne på 1-3 sekunder. Den højeste stigning er på 7 sekunder og kan til dels skyldes, at flere dokumenter matcher forespørgslen. Forøgelsen af tekstmængden fra 100 til 400

sider har således ikke givet anledning til et knæk på kurven over sammenhængen mellem tekstmængde og svartid. De fleste af udsvingene i svartid er så små, at de udelukkende kan skyldes afrundingsfejl. Forøgelsen af tekstmængden har således ikke betydet nævneværdigt større svartider. I forsøg 5 lagde vi alle 400 sider ind i tabellerne på én gang. Det gav mod forventning ikke nogen forskel i svartiderne. Det skyldes måske, at vi i forsøg 1-4 slettede de eksisterende indeks på tabellerne og oprettede nye, hver gang vi lagde mere tekst ind.

Afprøvningen har vist, at de mest omfattende søgninger giver svartider på under 1 minut - de ligger faktisk på ca 30 sekunder. De længste svartider forekommer i de svære søgninger, hvor der søges i en stor tekstmængde med 2 stk OG og 40 stk ELLER. Vi finder dette resultat særdeles tilfredsstillende. Det må specielt bemærkes, at søgning indenfor det store retsområde og søgning i hele tekstmængden giver samme svartider. Det skyldes den måde, forespørgslerne er implementeret på. Hele tekstmængden gennemsøges således også i de tilfælde, hvor søgningen er afgrænset til et retsområde.

De samme søgninger foretaget på forskellig tekstmængde - én lovtekst, et mindre retsområde, et større retsområde og alle lovtekster - har ikke medført markant forskel i svartiderne. For søgninger med kun to søgeord varierer svartiden fra 1 til 4 sekunder. For de øvrige søgninger er forskellen 1-2 sekunder og kan derfor skyldes afrundingsfejl. Det har været en positiv overraskelse for os, at medium søgninger og flere af de svære søgninger giver svartider på 4-8 sekunder.

Afprøvningen viser, at antallet af boolske operatorer og dermed antallet af søgeord har stor betydning for svartiderne. Mellem forespørgslerne med 1 boolsk operator og dem med 9 er der en forskel i svartiderne på ca 7 sekunder. Springet fra 9 boolske operatorer til 42 giver en virkelig markant forskel i svartiderne, 23 sekunder.

Brugen af joker-operatorer resulterer - ligesom mange boolske operatorer - i et stort antal søgeord. I et af afprøvningstilfældene resulterede tre ord med joker-operatorer, 'afskedige% og overenskomst% og berettige%', i godt 100 søgeord. Svartiden er 8 sekunder. Denne søgning blev udført både med nærhedsoperatoren 'indenfor samme lovtekst' og 'indenfor samme paragraf'. Svartiden er den samme i begge tilfælde. Sådan som vi har implementeret det, betyder det tilsyneladende ikke noget for svartiderne at bruge den stærkere nærhedsoprator 'indenfor samme paragraf'.

En vigtig faktor i vurderingen af de opnåede svartider er databasens størrelse. I den sammenhæng skal vi endvidere se på Edb-Karnovs pladsforbrug. De godt 400 sider tekst fra Virksomheds-Karnov fyldte knap 4 Mb, da vi fik dem fra Karnovs Forlag. I Edb-Karnov fylder de knap 19 Mb fordelt på teksttabellerne, søgeordstabellerne og deres indeks (se bilag 6). Brugen af en relationsdatabase har således ført til et ekstra pladsforbrug på næsten 400% i forhold til det, selve teksten fylder.

De godt 400 sider tekst, der er lagt ind i Edb-Karnov, giver anledning til 74.554 tupler i lovsøgeordstabellen og 110.064 tupler i notesøgeordstabellen. En søgning, der omfatter alle Edb-Karnovs lovtekster og noter betyder derfor gennemsøgning af knap 200.000 tupler. På den baggrund mener vi, afprøvningen har en vis realisme og giver nogle meget fine svartider, de fleste på 4-8 sekunder. I vurderingen afsvartiderne skal det endvidere tages i betragtning, at vi ikke har gjort meget ud af optimering.

Optimering

I det følgende vil vi beskrive nogle af de optimeringer, vi har foretaget for at opnå rimelige svartider. Vi vil ligeledes beskrive, hvor vi ser gode muligheder for yderligere optimeringer.

En afgørende faktor for at opnå korte svartider med en relationsdatabase er oprettelse

af passende indeks. Vi gennemgik SQL-forspørgslerne i vores programmer for at se hvilke attributer, der blev brugt som indgang til de forskellige tabeller, og hvilke data, der blev trukket ud af tabellerne. Ud fra denne gennemgang genererede vi de relevante indeks for hver tabel. Indeksenes betydningen viste sig tydeligt ved en fejl i vores afprøvningen. Vi havde glemt at oprette et indeks på lovoplysningstabellen. Det medførte, at svartiderne blev 3-4 gange længere for de tilfælde vi fik kørt, inden vi opdagede fejlen.

I Oracle er det muligt at specificere om et indeks skal pakkes eller ej. Pakkede indeks kan kun bruges som indgang til tabellerne; der kan ikke læses fra dem. De data, der skal læses, må altså hentes i selve tabellen. Indeks, der ikke er pakkede, kan der også læses fra. Med ikke-pakkede indeks kan opslaget i selve tabellen således undgåes. I starten brugte vi udelukkende pakkede indeks; men der kan opnåes meget store forbedringer af svartiderne ved at bruge ikke-pakkede indeks. Da vi ændrede søgeordstabellernes indeks til ikke-pakkede, betød det således i et par tilfælde, at svartiden forbedredes med en faktor 50.

En anden faktor, der er afgørende for svartiderne, er udformningen af SQL-forespørgslerne. Figur 4.20 viser et eksempel på en af Edb-Karnovs SQL-forespørgsler. I Edb-Karnov er alle SQL-forespørglerne bygget over den samme grundforespørgsel. Ex: Den SQL-forespørgsel, en søgning med nærhedsoperatoren 'indenfor samme paragraf' giver anledning til, består af grundforespørgslen med en tilføjelse, der sætter nærhedsoperatoren til 'indenfor samme paragraf'. De øvrige SQL-forespørgsler består ligeledes af grundforespørgslen med en eller anden tilføjelse. Der kan naturligvis vindes meget ved at udvikle en specifik SQL-forespørgsel til hver type søgning. For at få en indikation af, hvor meget der kan opnåes ved det, lavede vi en speciel SQL-forespørgsel for søgninger med nærheds-operatoren 'indenfor samme lovtekst'. Vi prøvede denne forespørgsel på et par af de omfattende afprøvningstilfælde. Resultatet var en forbedring af svartiden med en faktor 2 1/2, fra omkring 30 til 12 sekunder.

```
SELECT DISTINCT L.ART, L.AAR, L.NUMMER, L.NAVN
FROM LOVOPL L, LOVSOEGEORD S
WHERE (S.SOEGEORD LIKE 'afgift%')
AND S.LOVID = L.LOVID
INTERSECT
SELECT DISTINCT L.ART, L.AAR, L.NUMMER, L.NAVN
FROM LOVOPL L, LOVSOEGEORD S
WHERE (S.SOEGEORD LIKE 'straf%'
OR S.SOEGEORD LIKE 'ejendom%')
AND S.LOVID = L.LOVID
ORDER BY 2, 3
```

Figur 4.20. Eksempel på de SQL-forespørgsler, der genereres i Edb-Karnov. Brugeren har angivet forespørgslen 'afgift% og straf% eller ejendom%'. Søgningen omfatter alle lovtekster, og nærhedsoperatoren er 'indenfor samme lovtekst'. Det skal specielt bemærkes, at OG-operatoren omsættes til en INTERSECT.

Vi har foretaget en meget væsentlig optimering af SQL-forespørgslerne: Vi erstattede en *join* med en INTERSECT. I starten brugte vi en *join* til at repræsentere OG-operatoren. Det ændrede vi til at repræsentere OG-operatoren ved to delforespørgsler forbundet med INTERSECT. Hver delforespørgsel svarer til en separat søgning efter ét af de søgeord, der omgiver OG-operatoren. I et tilfælde betød det en forbedring af svartiden fra 2153 til 5 sekunder! Denne meget voldsomme forskel kan måske forklares ud fra de eksperimenter, Matos & Jalics (1989) har udført. Matos & Jalics finder, at PC-udgaven af Oracle er meget længe om at udføre *joins* af store tabeller. I et af deres eksperimenter er Oracle godt 6 timer om en *join*, som i Paradox udføres på 15 sekunder. Vi finder, at *joins* af store

tabeller også bør undgåes i Macintosh-udgaven af Oracle.

Det er også muligt at forbedre svartiderne ved at tune selve Oracle-databasesystemet. Det eneste, vi gjorde i den forbindelse, var at udvide sorteringsarealet fra 4K til 32K. Sorteringsarealet er den plads, Oracle bruger internt til sortering af tuplerne i resultatet af forespørgslerne eller i mellemresultater under udførelsen af forespørgslerne. De to mængder af tupler sorteres fx altid før en INTERSECT. Der kan gøres mere på dette område; men det har ikke været nødvendigt for at få Edb-Karnov til at opfylde de krav, vi havde sat til svartiderne.

Ovenfor har vi beskrevet de optimeringer, vi har foretaget. I det følgende vil vi pege på nogle af de optimeringsmuligheder, der stadig er tilbage og kan betyde markante forbedringer på svartiderne. Vi ser blandt andet følgende optimeringsmuligheder:

- I øjeblikket bygges alle SQL-forespørgsler op om samme grundforespørgsel. Det betyder i flere tilfælde, at der genereres unødigt komplicerede SQL-forespørgsler. SQL-forespørglserne skal således ikke bygges op om samme grundforespørgsel. Der skal i stedet udvikles en specifik SQL-forespørgsel til hver type søgning - en når der søges med nærhedsoperatoren 'indenfor samme lovtekst', en anden når søgningen afgrænses til et retsområde osv. Det vil give mulighed for at tune hver enkelt SQL-forespørgsel til netop den type søgninger, den skal behandle.
- I øjeblikket placeres søgeordene i SQL-forespørgslen i den rækkefølge, brugeren angiver dem i. Det vil reducere svartiderne for OG-operatoren, hvis søgeordene omplaceres sådan, at de søgeord, der forekommer færrest gange i lovteksterne, placeres først i SQL-forespørgslen. Ved at starte med de ord, der forekommer færrest gange, holdes den mængde af tekster, der skal gennemsøges, så lille som muligt. På denne måde minimeres den mængde tekster, der skal undersøges for forekomster af det andet søgeord, det tredje søgeord osv.
- Hvis der er mulighed for det, kan et eller flere af indeksene lægges ind i RAM, fremfor at ligge på disk.
- Selve Oracle-databasesystemet kan tunes yderligere med hensyn til buffere osv.
- Endelig kan svartiderne naturligvis forbedres ved at flytte Edb-Karnov til en større maskine. Da Oracle oprindeligt er udviklet til UNIX-maskiner, vil det være oplagt at vælge sådan en.

Konklusion på afprøvningen

Afprøvningen viser, at vi har opnået rimelige svartider med et fuldtekstsøgesystem baseret på en relationsdatabase. De fleste søgninger resulterede i ganske korte svartider, 4-8 sekunder. De tungeste søgninger resulterede i svartider på ca 30 sekunder, hvilket er væsentligt mindre end det minut, vi havde sat som maksimumsgrænse. Disse svartider er opnået uden, at vi har brugt særlig mange ressourcer på optimering. Der er således stadig mange uudnyttede optimeringsmuligheder. De godt 400 sider tekst, vi har lagt ind, fylder knap 19 Mb. Det svarer til et ekstra pladsforbrug på næsten 400% i forhold til det, teksten fylder som flad fil. Pladsforbruger er således stort, men efter vores mening ikke problematisk. De ca 4000 sider i hele Karnovs Lovsamling kan fx rigeligt være på en 300 Mb harddisk. På baggrund af vores prototype mener vi, det er realistisk at basere en edb-udgave af hele Karnovs Lovsamling på en relationsdatabase.

4.9 Sammenfatning

Igennem dette kapitel har vi udviklet en prototype på et juridisk fuldtekstsøgesystem, kaldet Edb-Karnov. Søgeteknikken i Edb-Karnov er boolsk søgning, hvor forespørgslerne formuleres med de boolske operatorer OG og ELLER. Derudover kan der benyttes to joker-operatorer og en enkelt nærhedsoperator i formuleringen af forespørgslerne.

Brugerne har behov for støtte i valget af søgeord. I Edb-Karnov består denne støtte af en on-line tesaurus. Herved adskiller Edb-Karnov sig fra de fleste fuldtekstsøgesystemer; de har kun sjældent en on-line tesaurus. Ordene i tesaurussen kan ikke blot bruges i forbindelse med fuldtekstsøgningen. Da der kan knyttes henvisninger til dem, kan ordene i tesaurussen også bruges som nøgleord. Henvisningerne kan være til såvel hele lovtekster som specifikke paragraffer, noter og notater. Med emneregistret er det muligt at afgrænse søgningerne til en udvalgt del af Edb-Karnovs tekster. Derved kan en del af de fremfundne, men irrelevante dokumenter undgåes; men der er også en risiko for at gå glip af relevante dokumenter.

Formålet med at implementere Edb-Karnov har været at konkretisere, udvikle og evaluere vores ideer, om faciliteter der tillader et søgesystem at udvikle sig over tid. I forbindelse med emneregistret er det resulteret i et søgesystem, der er meget sagsorienteret. Muligheden for, at den enkelte bruger kan foretage ændringer i emneregistret, betyder, at brugeren kan oprette sine egne sagsorienterede emner/registre. De kan fx oprettes i forbindelse med typiske sager, som juristen har en del af. Fordelen ved dem er, dels at lovteksterne opdeles efter brugerens behov, dels at de kan bruges til at afgrænse fremtidige søgninger indenfor lignende sager. Det dynamiske emneregister giver de fagligt kompetente brugere mulighed for at organisere lovteksterne på netop den måde, de finder mest overskuelig og hensigtsmæssig.

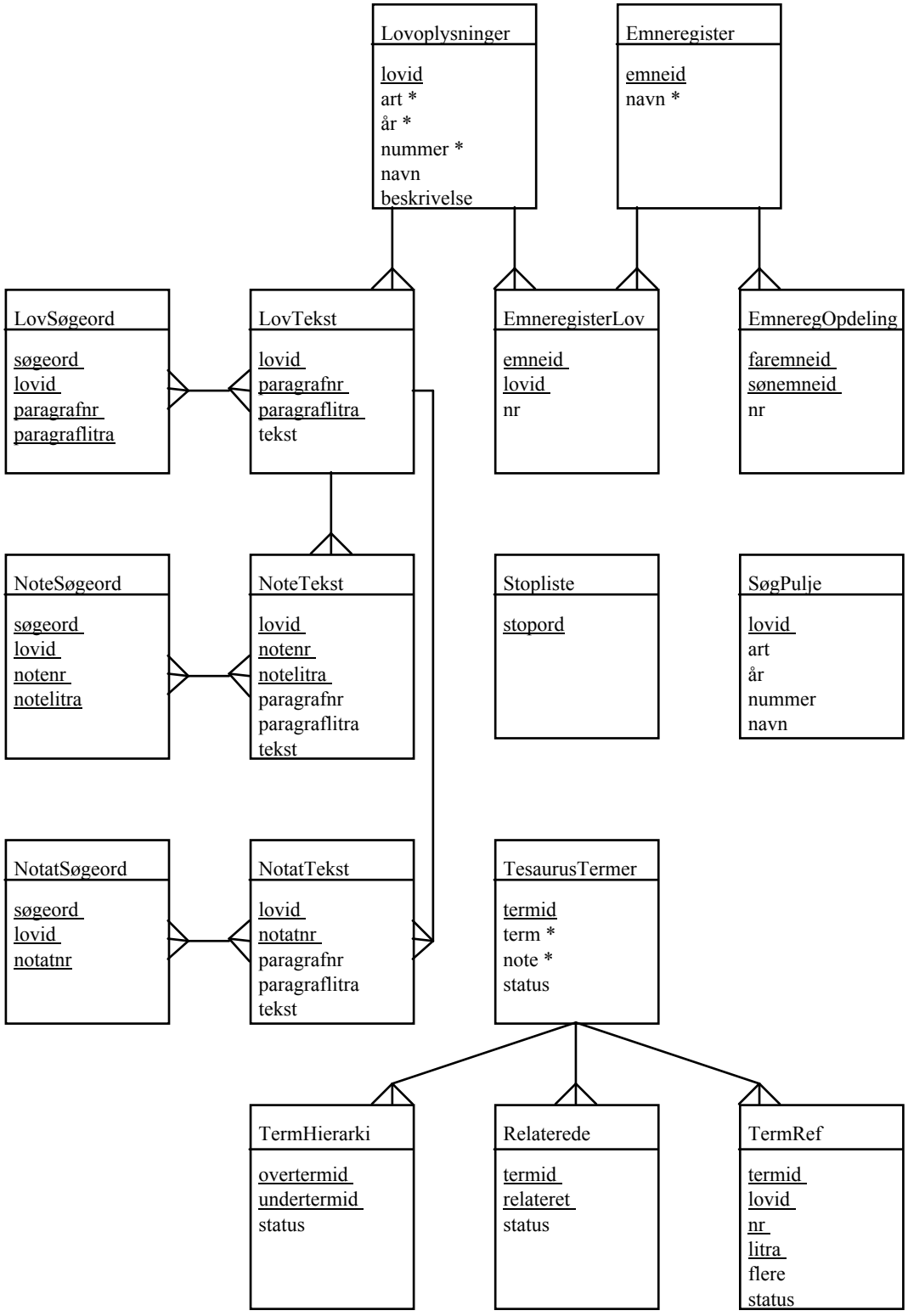
I tesaurussen findes ligeledes en redigeringsfunktion, der tillader at tesaurussen udvikles over tid. Her har vi givet mulighed for at arbejde med forslag til ændringer. Ændringerne i sprogbrug og begrebsdefinitioner er en glidende proces. Vi tror derfor, brugeren ofte vil have glæde af at kunne markere et ord eller en relation som et forslag. Derved huskes forslaget, og det kan senere vurderes, om forslaget var velvalgt eller skal slettes. Ligesom i emneregistret ser vi et behov for at have forskellige tesaurusser rettet mod hver deres område. Vi har kun lagt én grundtesaurus ind; men der er intet i vejen for, at brugeren tilføjer sine egne tesaurusser ved siden af.

Endelig støttes udvikling over tid af, at brugeren - i form af notater - kan føje sine egne tekster til dem, der allerede er i Edb-Karnov. Gennem notaterne er der mange muligheder for at integrere Edb-Karnov i juristens samlede arbejdssituation. Notaterne kan fx bruges til at etablere referencer mellem Edb-Karnov og sagerne i et journaliseringssystem. Vi har implementeret notater som en generel mulighed for at tilføje tekster. Derudover har vi behandlet bogmærker, som et eksempel på behovet for specielle typer notater rettet mod bestemte anvendelser.

Vi har udviklet Edb-Karnovs datamodel i takt med den gradvise udbygning af prototypens funktionalitet. Den færdige datamodel er gengivet i figur 4.21. I modellen systematiseres al tekst ud fra lovteksterne. Noter og notater er således altid knyttet til en lovtekst. Den gradvise udbygning af datamodellen har ikke givet anledning til problemer eller til mange ændringer i de allerede implementerede faciliteter. Specielt mener vi, notaterne er et eksempel på, hvor let datamodellen kan udvides med en ny type tekster. Ud fra det mener vi, det vil være let at føje fx domsafsigelser til Edb-Karnov på et senere tidspunkt.

Vi mener, den store fleksibilitet, der kendetegner relationsdatabaser, er den væsentligste årsag til, at udvidelserne af datamodellen er gået så let. En relationsdatabase har, som vi formodede, vist sig at være velegnet som grundlag for et søgesystem, der skal støtte udvikling over tid. Vi har intet empirisk grundlag for at udtale os om, hvorvidt relationsdatabaser er velegnede i forbindelse med fuldtekstsøgesystemer, der omfatter meget store tekstmængder. I Edb-Karnov har brugen af en relationsdatabase imidlertid været en succes. Vi vurderer således, at relationsdatabaser er en realistisk og fordelagtig mulighed for søgesystemer med en størrelse og funktionalitet, der minder om Edb-

Karnovs.



Figur 4.21. Den endelige datamodel for Edb-Karnov. Datamodellen er blevet udviklet gradvist hen igennem kapitlet; her vises den i sin helhed. Tabellernes primærnøgle er markeret ved understregning, alternative nøgler med stjerner.

5. Ajourføring - udgivelse af nye udgaver af Edb-Karnov

Forrige kapitel mundede ud i en prototype for det juridiske informationssøgesystem Edb-Karnov. I denne prototype har vi specielt fokuseret på faciliteter, der støtter udvikling over tid. Edb-Karnov giver således brugeren mulighed for at foretage løbende ændringer og tilføjelser i emneregistret, i tesaurussen og i form af egne notater. For at gøre implementeringen overkommelig afgrænsede vi os til at se på brugernes behov for at kunne foretage ændringer og tilføjelser i én udgave af Edb-Karnovs tekster. Da der løbende træder nye love, bekendtgørelser osv i kraft, skal teksterne i Edb-Karnov imidlertid også ajourføres. Ajourføringerne betyder ikke blot tilføjelse af lovtekster; de giver også anledning til ændringer i og tilføjelser til noterne, grundemneregistret og grundtesaurussen. Noterne ændres fx som følge af nye principielle domsafgørelser; grundemneregistret ændres fx som følge af, at nogle retsområder vokser og derfor med fordel kan underopdeles; og grundtesaurussen ændres fx som følge af ændringer i den juridiske sprogbrug.

Udgivelsen af nye udgaver af Edb-Karnov er et vigtigt aspekt af systemets udvikling over tid. Nye udgaver giver blandt andet anledning til overvejelser om, hvad der skal ske med ændrede og ophævede lovtekster. De skal efter vores mening ikke slettes fra systemet, men derimod overføres til en historisk database. En sådan historisk database kan let etableres som et nyt sæt tekst- og søgeordstabeller ved siden af dem, der indeholder de gældende lovtekster med tilhørende noter og notater. Det vil vi ikke behandle yderligere. I dette kapitel vil vi behandle mulighederne for at integrere de ændringer, brugerne foretager løbende, med udgivelsen af nye udgaver. Denne problemstilling er relevant i forbindelse med såvel notaterne som emneregistret og tesaurussen. Ex: Hvordan skal de rettelser og tilføjelser, brugeren har foretaget løbende, overføres til en ajourført udgave af grundtesaurussen?

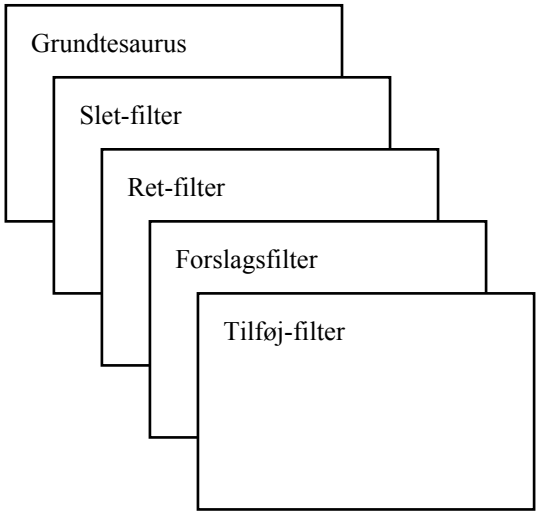
Vi vil gennem hele kapitlet bruge tesaurussen som eksempel og grundlag for vores overvejelser. I kapitlets første afsnit behandler vi, hvordan grundtesaurussen kan adskilles fra de ændringer, brugeren foretager ved hjælp af redigeringsfunktionerne. Derefter følger kapitlets hovedafsnit. Her beskriver vi vores forslag til, hvordan løbende ændringer og nye udgaver af tesaurussen kan integreres. Kapitlet slutter med en sammenfatning, hvor vi også ser på mulighederne for at bruge vores forslag i forbindelse med emneregistret og notaterne.

5.1 Mulighed for at genskabe grundtesaurussen

I tesaurussen kan brugeren tilføje ord og relationer mellem ord samt slette og rette i såvel grundtesaurussen som egne tilføjelser. Indtil nu er alle ændringer af tesaurussen blevet behandlet ens, uanset om der var tale om ændringer i grundtesaurussen eller i brugerens tilføjelser til den. Det er imidlertid uhensigtsmæssigt af to grunde: Det er for det første ikke muligt at genskabe grundtesaurussen automatisk, når der én gang er blevet ændret i den. For det andet er brugerens ændringer i grundtesaurussen ikke tilgængelige - det er kun tesaurussens øjeblikkelige udseende, der er tilgængeligt. Direkte adgang til ændringerne af grundtesaurussen er nødvendig, når en ny grundtesaurus skal installeres. Første skridt i integrationen af brugerens løbende ændringer og udgivelsen af nye udgaver af grundtesaurussen er derfor at adskille grundtesaurussen og brugerens ændringer.

Vi vil foretage denne adskillelse ved at betragte tesaurussen som en grundtesaurus samt fire filtre: Et slet-filter, et ret-filter, et forslagsfilter og et tilføj-filter (se figur 5.1).

Med denne opdeling kan grundtesaurussen genskabes permanent ved at slette de fire filtre. Grundtesaurussen kan også genskabes midlertidigt ved at læse grundtesaurussen, men springe filtrene over. En facilitet til midlertidigt at genskabe grundtesaurussen er særlig relevant i de tilfælde, hvor brugeren har foretaget ændringer og senere installeret en ny udgave af grundtesaurussen. Her får brugeren midlertidig adgang til den nye grundtesaurus uden at give afkald på de foretagne ændringer.



Figur 5.1. Tesaurussen organiseret som en grundtesaurus og fire filtre. Slet-filteret indeholder de ord og relationer mellem ord, som brugeren har slettet fra grundtesaurussen. Ret-filteret indeholder de ændringer i stavemåde eller lignende, som brugeren har foretaget på grundtesaurussens ord; dette filter bruges ikke i forbindelse med relationerne mellem ordene. Forslagsfilteret indeholder de ord og relationer mellem ord, der har status af forslag. Tilføj-filteret indeholder de nye ord og relationer mellem ord, som brugeren har føjet til grundtesaurussen.

Implementering af tesaurussen i denne form kræver ingen ændringer i datamodellen. De tabeller, der udgør tesaurussen, indeholder alle en status-attribut, som i øjeblikket bruges til at skille forslagene fra de godkendte ord og relationer mellem ord. Denne attribut kan i stedet bruges til angivelse af, om den pågældende tupel hører til grundtesaurussen eller til et af de fire filtre. Attributen kan således have fem forskellige værdier:

'G' hvis tuplen tilhører grundtesaurussen,
'S' hvis tuplen tilhører slet-filteret,
'R' hvis tuplen tilhører ret-filteret,
'F' hvis tuplen tilhører forslagsfilteret og
'T' hvis tuplen tilhører tilføj-filteret.

De af Edb-Karnovs funktioner, der læser fra tesaurussens tabeller, skal tage højde for, at en tupel fra grundtesaurussen muligvis overskrives af en fra slet-, ret- eller forslagsfilteret. En tupel i slet- eller ret-filteret er altid knyttet til den af grundtesaurussens tupler, der modificeres. I tabellerne afspejles denne tilknytning ved, at de to tupler er ens pånær status-attributen. Tabellernes nøgler udvides således med status-attributen. I det omfang, tuplerne i forslagsfilteret modificerer indholdet af grundtesaurussen, vil også de give anledning til to tupler i tabellerne, der er ens pånær status-attributen. Derudover kan en tupel forekomme i forslagsfilteret eller grundtesaurussen uden at forekomme andre steder, og endelig forekommer en tupel i tilføj-filteret altid kun der.

5.2 Integration af løbende ændringer og nye udgaver

Når en ny udgave af grundtesaurussen udkommer, skal den kunne lægges ind i Edb-

Karnov. Spørgsmålet er, hvordan det kan gøres samtidig med, at de ændringer, brugeren har foretaget løbende, bevares. Integrationen af de løbende ændringer og den nye udgave af grundtesaurussen kan ske ved, at de løbende ændringer flyttes fra den gamle til den nye udgave af grundtesaurussen. Brugerne vil imidlertid betragte ajourføringen af tesaurussen på en anden måde: De vil se på, hvilke ændringer ajourføringen af grundtesaurussen introducerer i den tesaurus, de kender og arbejder med. Da ajourføringen ikke kan foregå helt automatisk, men i et vist omfang vil involvere brugerne, skal den efter vores mening foregå sådan, som de betragter den. Brugerne skal involveres i de tilfælde, hvor den nye grundtesaurus og deres egne ændringer er i konflikt; her er det brugernes valg, om den nye grundtesaurus skal overskrive deres ændringer eller omvendt. I Edb-Karnov skal ajourføringen altså ske ved, at ændringerne i grundtesaurussen indføres i den tesaurus, brugeren arbejder med (den anden mulighed var at flytte brugerens ændringer fra den gamle til den nye grundtesaurus).

For udbyderen af Edb-Karnov er den primære opgave ajourføringen af lovtekster, noter, grundemneregister og grundtesaurus. Angivelsen af ændringerne i grundtesaurussen er således en opgave, der vil være opmærksomhed om og sat tid af til. For brugeren er det søgningerne, der er begrundelsen for at bruge Edb-Karnov, så her er det vigtigt, at redigeringsfunktionerne og de andre funktioner, der skal støtte søgningerne, er enkle og lette at gå til. Hvis størstedelen af arbejdet med ajourføringen kan placeres hos udbyderen, vil det endvidere have den fordel, at dette arbejde kun skal udføres én gang. Den del af arbejdet med ajourføringen, som brugeren skal udføre, vil derimod belaste mange mennesker, da den skal udføres for hver installation af Edb-Karnov.

På den baggrund opnåes en væsentlig fordel ved at vælge at overføre ændringerne i grundtesaurussen til den tesaurus, brugeren arbejder med: Hele den detaljerede angivelse af ændringerne i grundtesaurussen skal foretages af udbyderen, mens brugen af Edb-Karnov - specielt brugen af redigeringsfunktionerne - ikke påvirkes. Hvis man i stedet valgte at flytte brugernes ændringer fra den gamle til den nye grundtesaurus, ville flere og mere komplicerede redigeringsfunktioner være nødvendige. Det skyldes, at flytningen kun kan foretages, hvis den sammenhæng, de enkelte slet, ret og tilføj indgår i, er udtrykt eksplicit. Ex: Hvis grundtesaurussen efter brugerens mening er for detaljeret på et eller andet område, kan brugeren slå flere af tesaurussens ord sammen til ét og erstatte alle relationer, der involverer disse ord, med relationer til og fra det nye ord. Efter en sådan sammensmeltning skal også tilføjelse af nye relationer, der involverer de sammensmeltede ord, føre til relationer til og fra det nye ord. Det er imidlertid umuligt at tilføje sådanne relationer under ajourføringen af grundtesaurussen, hvis brugerens ændringer kun består af isolerede slet, ret og tilføj. Det skyldes, at der ikke er angivet nogen sammenhæng mellem de slettede ord og det nye ord.

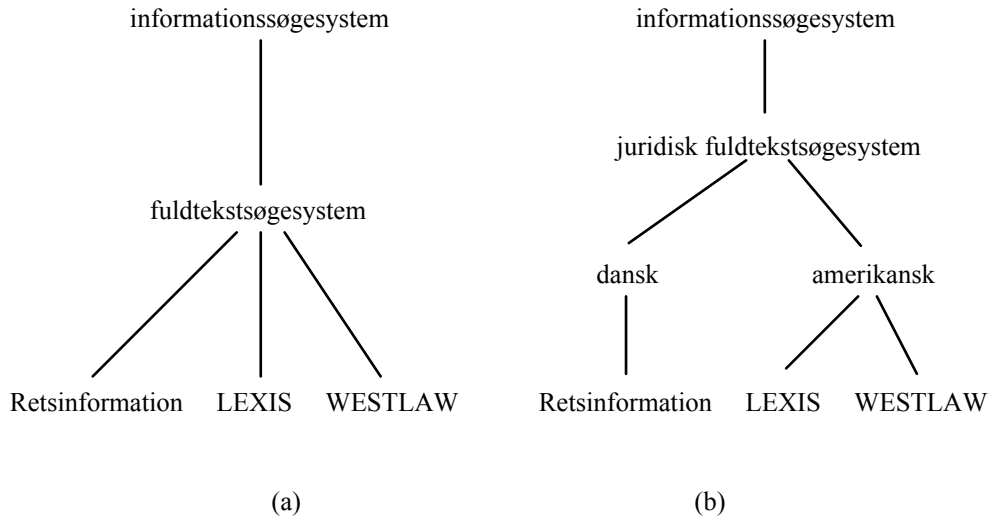
Udbyderen skal ikke beskrive den nye udgave af grundtesaurussen i sig selv. Den nye udgave skal udelukkende beskrives som en række ændringer i forhold til den gamle udgave. Disse ændringer skal beskrives på en måde, der muliggør maskinel behandling. Med det for øje finder vi det fordelagtigt at betragte tesaurussen som en graf, hvor ordene er grafens knuder og relationerne mellem ordene dens kanter. Ændringer i tesaurussen svarer således til ændringer i en graf. Enhver ændring i en graf kan foretages, hvis det er muligt at slette, rette og tilføje kanter såvel som knuder. Enhver ændring i grundtesaurussen kan således beskrives, hvis det er muligt at slette, rette og tilføje ord samt slette og tilføje relationer mellem ord. Muligheden for at rette relationer mellem ord er overflødig her, da der ikke er knyttet nogen oplysninger, fx en vægt, til relationerne. Enhver ændring af grundtesaurussen kan således beskrives ved hjælp af fem grundfunktioner:

Slet(a), der sletter ordet a fra grundtesaurussen.

Ret(a, b), der erstatter ordet a i grundtesaurussen med b.
Tilføj(a), der tilføjer ordet a til grundtesaurussen.
Forbind(a, b), der skaber en relation mellem ordene a og b.
Afbryd(a, b), der fjerner relationen mellem ordene a og b.

Forbind og *afbryd* skal i praksis også have en parameter, der angiver den involverede relation; den undlader vi imidlertid her af hensyn til overskueligheden. De fem grundfunktioner er lette at implementere; men hver af dem resulterer i så lille en ændring, at de ikke giver noget overblik over de foretagne ændringer. Grundfunktionerne vil således typisk blive arrangeret i grupper, der hver udgør en ændring på et abstraktionsniveau, der svarer til udbyderens og brugernes.

Da ajourføringen så vidt muligt bør foregå på udbyderens og brugernes abstraktionsniveau, vælger vi at organisere ajourføringen omkring sekvenser af grundfunktioner. Disse sekvenser defineres af udbyderen og består af et navn, en beskrivelse og en krop. Det er kroppen, der er selve sekvensen af de fem grundfunktioner. Vi kan umiddelbart forestille os tre sekvenser af grundfunktioner, som vil forekomme hyppigt: En flytsekvens, der blot består af en *afbryd* efterfulgt af en *forbind*, samt en opsplit- og en sammensmelt-sekvens:



Figur 5.2. Illustration af sekvenserne opsplit og sammensmelt. Ordet 'fuldtekstsøgesystem' i (a) er i (b) opsplittet i 'juridisk fuldtekstsøgesystem', 'dansk' og 'amerikansk', desuden er relationerne til og fra 'fuldtekstsøgesystem' fordelt på de tre ord, det opsplittes i. Definitionen af denne opsplit-sekvens er vist i figur 5.3. Sammensmelt er det modsatte af opsplit, altså ændringen fra (b) til (a).

Når udbyderen skal angive de ændringer, der udgør ajourføringen af tesaurussen, kan det ske ved at gruppere og indtaste grundfunktionerne som illustreret i figur 5.3. Det fører imidlertid til en ret omstændelig arbejdsproces med mange muligheder for fejl i angivelsen af ændringerne. En væsentligt bedre løsning er at basere denne arbejdsproces på en grafisk brugergrænseflade. Her kan udbyderen manipulere med tesaurussen på skærmen ved at markere de ord og relationer mellem ord, der skal ændres, og derefter klikke på den ønskede grundfunktion. Resultatet af disse manipulationer er, dels at tesaurussen ændres på skærmen, dels at de udførte grundfunktioner lagres i en ajourføringsfil.

opsplit:

beskrivelse: 'fuldtekstsøgesystem' opsplittes i 'juridisk fuldtekstsøgesystem', 'dansk' og 'amerikansk'. Opsplitningen omfatter dels en præcisering af, at der kun er tale om fuldtekstsøgesystemer indenfor det juridiske område, dels en geografisk opdeling af systemerne.

ret(fuldtekstsøgesystem, juridisk fuldtekstsøgesystem)
tilføj(dansk)
forbind(juridisk fuldtekstsøgesystem, dansk)
afbryd(juridisk fuldtekstsøgesystem, Retsinformation)
forbind(dansk, Retsinformation)
tilføj(amerikansk)
forbind(juridisk fuldtekstsøgesystem, amerikansk)
afbryd(juridisk fuldtekstsøgesystem, LEXIS)
forbind(amerikansk, LEXIS)
afbryd(juridisk fuldtekstsøgesystem, WESTLAW)
forbind(amerikansk, WESTLAW).

Figur 5.3. Definitionen af den opsplit-sekvens, der hører til figur 5.2. Funtionen består af tre dele: Et navn, en beskrivelse og en krop.

For brugeren af Edb-Karnov finder ajourføringen sted ved, at de sekvenser, udbyderen har defineret, udføres én af gangen. Udførelsen af hver sekvens af grundfunktioner foregår så vidt muligt automatisk. Brugeren involveres kun i de tilfælde, hvor systemet ikke kan afgøre, hvilke omplaceringer der skal foretages. Det sker, når ajourføringen af grundtesaurussen er i konflikt med en af de ændringer, brugeren har udført. En sådan konflikt indtræffer fx, hvis brugeren har etableret en relation mellem to ord, og ajourføringen af grundtesaurussen foreskriver, at det ene af dem skal slettes.

I visse tilfælde kan det også være relevant at have mulighed for at foretage ajourføringen helt manuelt. Det kan foregå ved, at Edb-Karnov præsenterer brugeren for hver eneste ændring, som derefter kan godkendes eller forkastes. Manuel ajourføring er fx relevant, hvis brugeren ikke ønsker nogen ændringer i sin tesaurus, men alligevel vil installere den nye udgave af grundtesaurussen for at være klar til næste gang, der kommer en ny udgave. I de tilfælde opnåes det ønskede ved at forkaste alle de ændringer, der indgår i ajourføringen. Derved indføres ændringerne i grundtesaurussen, hvorefter de straks ophæves igen ved hjælp af filtrene. Ajourføringen skal jo altid føre til, at den nye grundtesaurus helt erstatter den gamle - ellers kan tesaurussen ikke ajourføres næste gang, der kommer en ny udgave.

For at konkretisere diskussionen af ajourføringsfaciliteterne vil vi i den sidste del af dette afsnit se på, hvordan Edb-Karnov kan udføre ajourføringen. Ved udførelsen checker Edb-Karnov, om alle grundfunktionerne i den næste af de sekvenser, udbyderen har defineret, kan udføres uden brugerindblanding. Hvis det er tilfældet, udføres ændringerne automatisk. Ellers udskrives den beskrivelse, der indgår i sekvensen, og derefter præsenteres brugeren for de grundfunktioner, der udgør sekvensen. Præsentationen kan ske ved at udskrive grundfunktionerne i et vindue. En mere avanceret mulighed er at bruge en grafisk brugergrænseflade og angive ændringerne grafisk på den eksisterende tesaurus samt fremhæve dem ved hjælp af farver eller lignende. I begge tilfælde skal brugeren have mulighed for at bladre eller på anden måde orientere sig i alle de involverede ændringer. Derefter skal hver enkelt grundfunktion godkendes eller forkastes. Brugeren bør også have mulighed for selv at foretage redigeringer; sådanne redigeringer påvirker naturligvis ikke grundtesaurussen, men kun filtrene.

Vi vil ikke gennemgå alle de fem grundfunktioner, men blot som eksempel se på

forbind. *Forbind* bruges under ajourføringen af grundtesaurussen, når udbyderen vil etablere en relation mellem to begreber, der allerede findes i grundtesaurussen. I det følgende vil vi se på, hvordan udførelsen af *forbind* forløber. Udførelsen af *forbind*(a, b) afhænger dels af, om ordene a og b forekommer i filtrene, dels af om filtrene siger noget om relationen mellem a og b. I det følgende benævnes relationen mellem a og b: (a, b). (a, b) kan ikke forekomme i ret-filteret, da relationer ikke kan rettes. (a, b) kan heller ikke forekomme i slet-filteret, da det ville betyde, at relationen også fandtes i grundtesaurussen. Relationen (a, b) findes således enten i tilføj-filteret, i forslagsfilteret, eller den findes hverken i grundtesaurussen eller filtrene.

Hvis (a, b) findes i tilføj- eller forslagsfilteret, er brugeren kommet udbyderens ajourføring i forkøbet. I dette tilfælde flyttes (a, b) fra filteret til grundtesaurussen ved, at a og b forbindes i grundtesaurussen, mens forbindelsen i tilføj- eller forslagsfilteret fjernes.

Hvis (a, b) hverken findes i grundtesaurussen eller nogle af filtrene, vil ajourføringen introducere en relation mellem to ord, der hidtil ikke har stået i relation til hinanden. I dette tilfælde afhænger ajourføringen af, om a eller b forekommer i filtrene. Da ajourføringen er baseret på grundtesaurussen, vil såvel a som b forekomme i grundtesaurussen; men det er muligt, at de desuden forekommer i slet-, ret- eller forslagsfilteret. Hvis enten a eller b tilhører slet-filteret, har brugeren derved også markeret, at en eventuel forbindelse mellem a og b er uden interesse. Resultatet bliver derfor, at a og b forbindes i grundtesaurussen, og at den derved skabte relation straks ophæves igen ved også at blive føjet til slet-filteret. I de øvrige tilfælde, dvs når a og b forekommer i grundtesaurussen og eventuelt også i ret- eller forslagsfilteret, udvides tesaurussen med (a, b), ved at a og b forbindes i grundtesaurussen.

5.3 Sammenfatning

I dette kapitel har vi vist, hvordan brugerens løbende ændringer af tesaurussen kan integreres med udgivelsen af nye udgaver af grundtesaurussen. Resultatet er, at grundtesaurussen kan ajourføres uden, at brugerens egne ændringer går tabt eller skal flyttes manuelt fra den gamle til den nye grundtesaurus. Kapitlet er også en understregning af, at ajourføringen er en kompleks problemstilling. Vi har kun beskæftiget os med, hvordan de løbende ændringer og de nye udgaver kan integreres; men det kan fx også være relevant at tidsstemple de forskellige ændringer. Det vil give mulighed for at genskabe tesaurussen, som den så ud pr den og den dato. Ajourføringens kompleksitet kommer også til udtryk ved, at den ikke kan foregå helt automatisk. Det skyldes, at Edb-Karnov hverken kan eller skal afgøre, om grundtesaurussen skal overskrive brugerens ændringer eller omvendt i de tilfælde, hvor der er konflikt mellem dem. Det skal pointeres, at de konflikter, der opdages på denne måde, kun er dem, der kommer til udtryk som konflikter mellem knuder/kanter i den grafiske fremstilling af tesaurussen. Betydnings- eller indholdsmæssige konflikter mellem brugerens ændringer og ajourføringen vil langtfra altid, komme til udtryk på denne måde.

Antallet af sådanne konflikter afhænger helt af omfanget af de redigeringer, brugeren har foretaget i tesaurussen. Hvis brugeren ikke har anvendt redigeringsfunktionerne, er der naturligvis ingen konflikter, og ajourføringen sker helt automatisk. Ajourføringen foregår også helt automatisk i de tilfælde, hvor brugerens og udbyderens ændringer ikke berører de samme ord eller relationer mellem ord. Der opstår heller ingen konflikter, så længe alle de ændringer, der berører det samme ord eller den samme relation, er tilføjelser. Det er, når sletninger og rettelser støder sammen med andre ændringer (tilføjelser, sletninger eller rettelser), at der er risiko for konflikter. Vi kan ikke udtale os om antallet af sådanne konflikter; men hvis brugeren synes, de er svære at afgøre, kan afgørelsen meget let

udskydes: Hvis brugeren konsekvent fastholder sine egne ændringer, vil såvel disse ændringer som dem, den nye udgave af grundtesaurussen introducerer, være tilgængelige. Det sikres af, at grundtesaurussen altid kan genskabes.

Ajourføringen af emneregistret kan behandles på fuldstændig samme måde som tesaurussen. Der skal tilføjes en status-attribut til de berørte tabeller, da en sådan ikke findes i forvejen. Derudover er forslagsfilteret overflødigt, da det, sådan som vi har implementeret Edb-Karnovs emneregister, ikke er muligt at benytte forslag i emneregistret. Udover disse to punkter er der ingen forskel på ajourføringen af emneregistret og ajourføringen af tesaurussen.

Ajourføringen af notaterne er lidt anderledes; men princippet er stadig det samme som for tesaurussen. I forbindelse med notaterne udgør lovteksterne det, der svarer til grundtesaurussen, og der er kun ét filter: Tilføj-filteret. For tesaurussens vedkommende beskrev udbyderen ændringerne ved hjælp af fem grundfunktioner. For notaternes vedkommende vil en tabel nok være et mere naturligt valg. Denne tabel skal fastlægge, hvilke paragraffer der erstatter hvilke. For hver paragraf i de ændrede lovtekster skal tabellen angive den paragraf, der erstatter den ændrede paragraf. Paragraffer i lovtekster, som udgår og ikke erstattes af nye, får en særlig markering. De paragraffer, hvor det af andre grunde ikke er muligt at udpege en erstattende paragraf, får ligeledes en særlig markering. Herefter foregår ajourføringen ved at flytte notaterne fra de gamle lovtekster til de nye. Princippet er det samme som for tesaurussen.

6. Demonstration og evaluering af Edb-Karnov

I dette kapitel afslutter vi behandlingen af vores case med en demonstration af Edb-Karnov for såvel læseren som nogle indbudte personer. Demonstrationen har for det første til formål at give et overblik over prototypen og dens funktioner. For det andet udgør demonstrationen en evaluering af Edb-Karnov. Denne evaluering består af de indbudte personers kommentarer til og kritik af Edb-Karnov. Demonstrationerne forløb ved, at vi gennemgik et eksempel. Vi lagde vægt på, at eksemplet skulle være intuitivt nemt at forstå og præsentere Edb-Karnovs væsentligste funktioner. De funktioner vi ville præsentere var:

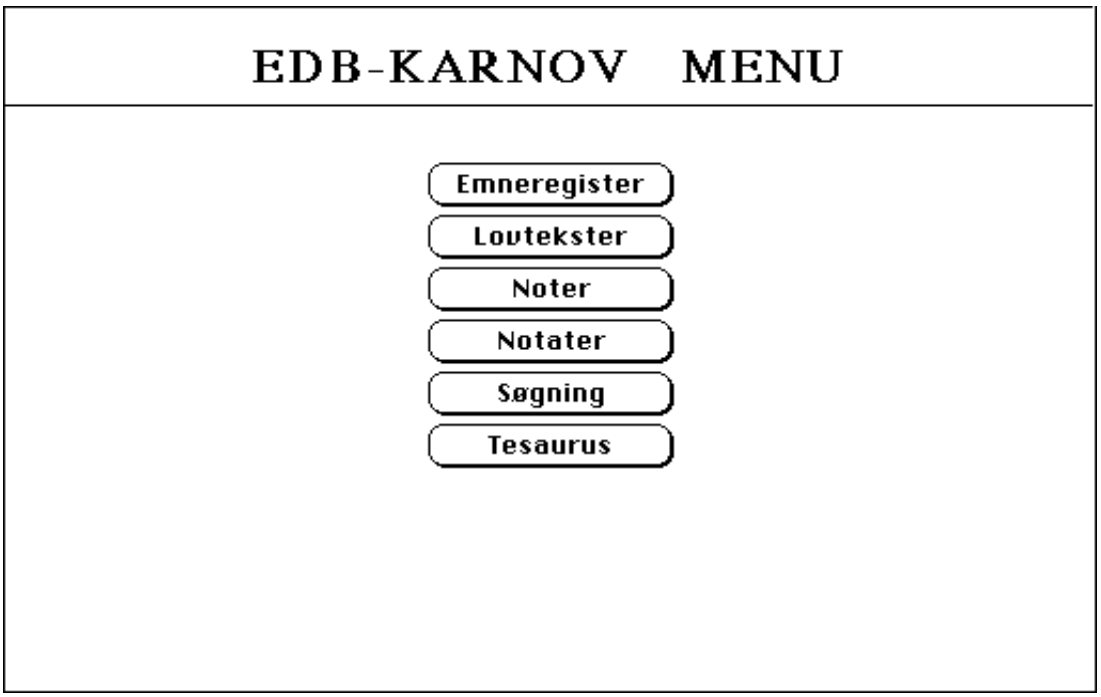
- Søgning.
- Tesaurussen: Hvorledes kan tesaurussen støtte valget af søgeord, og hvordan kan brugeren tilføje nye ord.
- Emneregistret: Hvorledes kan emneregistret bruges til at begrænse søgningens omfang til et emne, og hvordan kan et nyt emne, 'min sag', oprettes.
- Oprettelse af et personligt notat.
- Præsentation af lovtekst.
- Eksport af tekst fra Edb-Karnov.

Vi afholdt tre demonstrationer af Edb-Karnov. Den første demonstration var for direktør Jens Peter Nielsen, Karnovs Forlag. Han repræsenterer en potentiel udbyder af en eventuel færdigudviklet udgave af Edb-Karnov. Karnovs Forlag har lang tids erfaring i at støtte jurister i sagsbehandlingen og påtænker desuden selv at udvikle en elektronisk udgave af Karnovs Lovsamling. Jens Peter Nielsen har således en god baggrund for at vurdere, om Edb-Karnov vil kunne støtte jurister i deres arbejde. Da vi demonstrerede Edb-Karnov for Jens Peter Nielsen var der desuden to personer til stede fra den koncern, der ejer Karnovs Forlag; direktørerne Svend Erik Skydt og Kresten Bager.

Edb-Karnov blev endvidere demonstreret for Per Sjøqvist, den ene af de to jurister vi interviewede i forbindelse med vores indkredsning af juridisk sagsbehandling. Per Sjøqvist repræsenterer en potentiel bruger af Edb-Karnov. Vi har ikke demonstreret Edb-Karnov for Dorthe la Cour, den anden jurist vi interviewede. Det skyldes, at hun er på barselsorlov.

Den sidste demonstration, vi foretog, var for Peter Ingwersen, Biblioteksskolen. Han er en anerkendt forsker indenfor informationssøgning og særlig interessant for os, fordi han kan at se ud over vores case og vurdere Edb-Karnov i forhold til andre søgesystemer.

Deltagerne i demonstrationen blev ikke instrueret i, at der var noget, de specielt skulle kommentere eller lægge mærke til ved Edb-Karnov. Under demonstrationerne kom deltagerne med spørgsmål, kritik og kommentarer, og det førte til forskellige diskussioner om Edb-Karnov. I dette kapitel vil vi give en samlet præsentation af Edb-Karnov og referere den kritik og de kommentarer, Edb-Karnov fik under demonstrationerne.



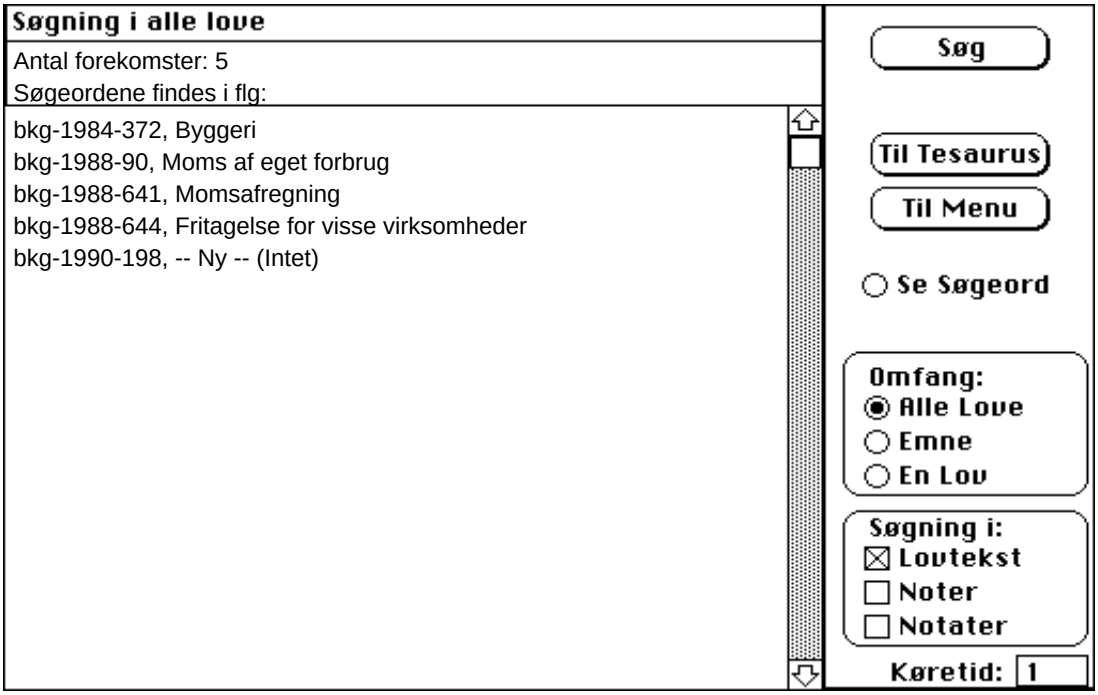
Figur 6.1.Menu-skærmbilledet. Dette skærmbilledet er indgangen til Edb-Karnov og viser de overordnede funktioner, der er til rådighed.

6.1 Demonstration

Det eksempel, demonstrationen er baseret på, er følgende:

> Vi skal til at starte på en sag, hvor vi som det første skal finde grundlaget for momsberegningen.

Det følgende bliver en præsentation af, hvorledes grundlaget for momsberegningen kan findes ved hjælp af Edb-Karnov. Søgningen starter med menu-skærmbilledet (figur 6.1), hvor der klikkes på Søgning for at komme til søge-skærmbilledet.

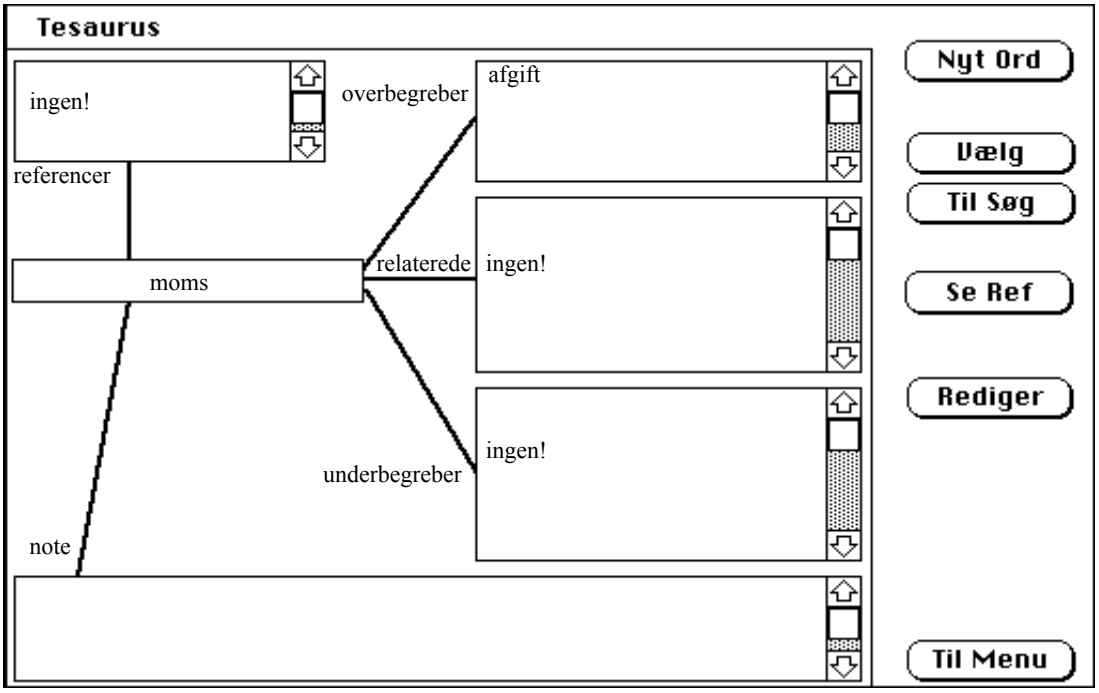


Figur 6.2. Figuren viser resultatet af søgningen med det ene søgeord 'moms%'. Søgningen omfatter alle love, og der søges kun i lovtekster.

Da vi skal finde grundlaget for momsberegningen, er det naturligt at vælge 'moms%' som søgeord, hvor % er en joker-operator, der betyder, at der søges på alle ord, der starter med 'moms'. Samtidig ved vi, at grundlaget for momsberegningen skal findes i en lovtekst. Derfor starter vi med at begrænse søgningen til kun at omfatte lovtekster. Resultatet af søgningen er fem bekendtgørelser (se figur 6.2). Grundlaget for momsberegningen kan ikke være en bekendtgørelse eller et cirkulære, så resultatet er ikke det ønskede. Resultatet tyder på, at ordet 'moms' ikke forekommer i den lovtekst, hvor grundlaget for momsberegningen fastlægges, og derfor ikke er et velegnet søgeord.

Både Jens Peter Nielsen og Per Sjøqvist påpeger i denne forbindelse, at det netop er det store problem ved fuldtekstsøgning, at søgeordene ikke forekommer i teksten. I et nøgleordsbaseret søgesystem mener de, det vil være muligt at søge på moms og hurtigt blive ledt hen til de relevante dokumenter og paragraffer. De understreger, at brugerne skal/vil ledes specifikt til de relevante paragraffer. Heroverfor fremhæver vi, at nøgleord rammes af forældelse og er følsomme overfor forskellige personers forskellige synsvinkler på teksterne. I Edb-Karnov har vi i stedet givet mulighed for at reducere problemet, med at søgeordene ikke står i teksten, ved at indføre en tesaurus. Tesaurussen skal støtte brugeren i valget af søgeord.

Vi klikker derfor på Til Tesaurus for at se, om tesaurussen kan give os en idé til et andet valg af søgeord.

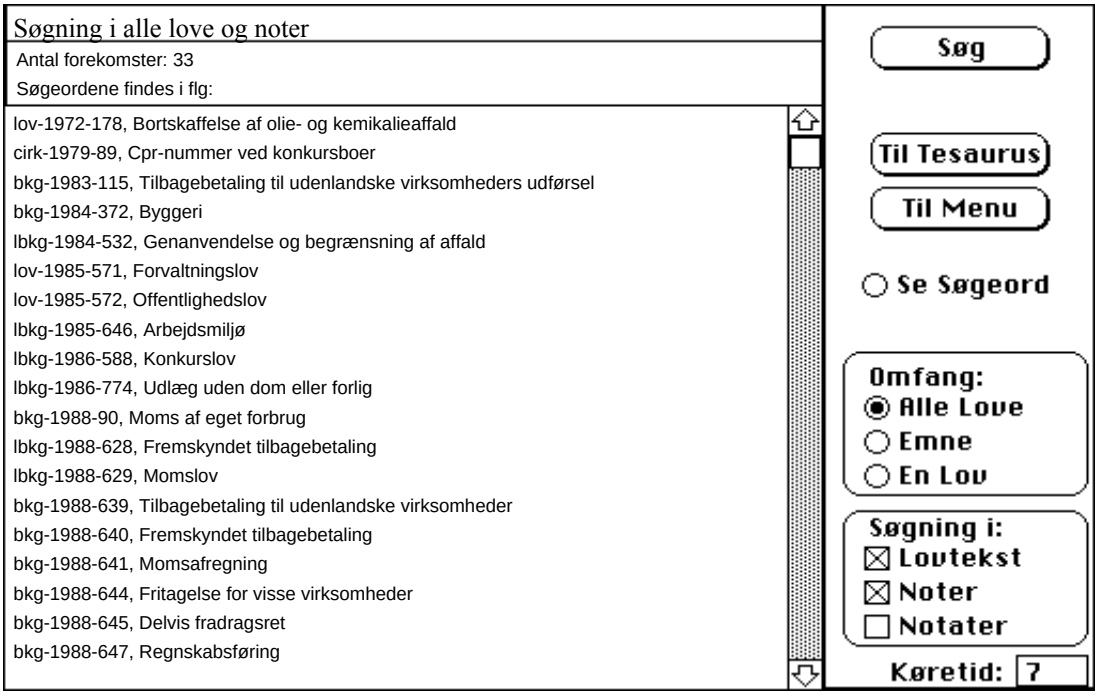


Figur 6.3. Tesaurusen viser, at 'moms' kun har et overbegreb, 'afgift', men ingen relaterede begreber og heller ingen underbegreber.

I tesaurussen er vi interesserede i at se, hvilke relationer ordet moms har til andre begreber. Vi klikker derfor på Nyt Ord, og indtaster 'moms', hvorefter moms og dets relationer til andre begreber fremkommer, som det ses på figur 6.3. På denne figur ses, at moms kun har ét overbegreb, afgift, og hverken har relaterede begreber eller underbegreber.

Jens Peter Nielsen og Per Sjøqvist spørger, om de relaterede begreber automatisk føjes til forespørgslen, hvis der er nogle. Det gør de ikke; men det skal efter Jens Peter Nielsens og Per Sjøqvists mening være en mulighed. De oplever begge, de relaterede begreber anderledes end resten af tesaurussen. Relaterede begreber bruges til at gøre forespørgslerne mindre følsomme overfor variationer i ordvalget i dokumenterne; der vil således blive føjet relaterede begreber til de fleste søgeord. Over- og underbegreber bruges derimod til at finde de søgeord, der bedst beskriver den utilstrækkelige viden. Denne adskillelse kunne tyde på, at de relaterede begrebers placering i tesaurussen med fordel kunne ændres. De kunne eventuelt gives en helt selvstændig position som synonymordbog ved siden af tesaurussen. Endelig bemærker Per Sjøqvist, at sagregistret fra Virksomheds-Karnov ikke er den mest velegnede tesaurus.

Da moms kun har overbegrebet afgift, beslutter vi at prøve at søge på afgift, selvom et overbegreb formodentlig vil betyde, at flere irrelevante dokumenter fremfindes. Vi klikker på Til Søg for at fortsætte søgningen i søge-skærmbilledet.



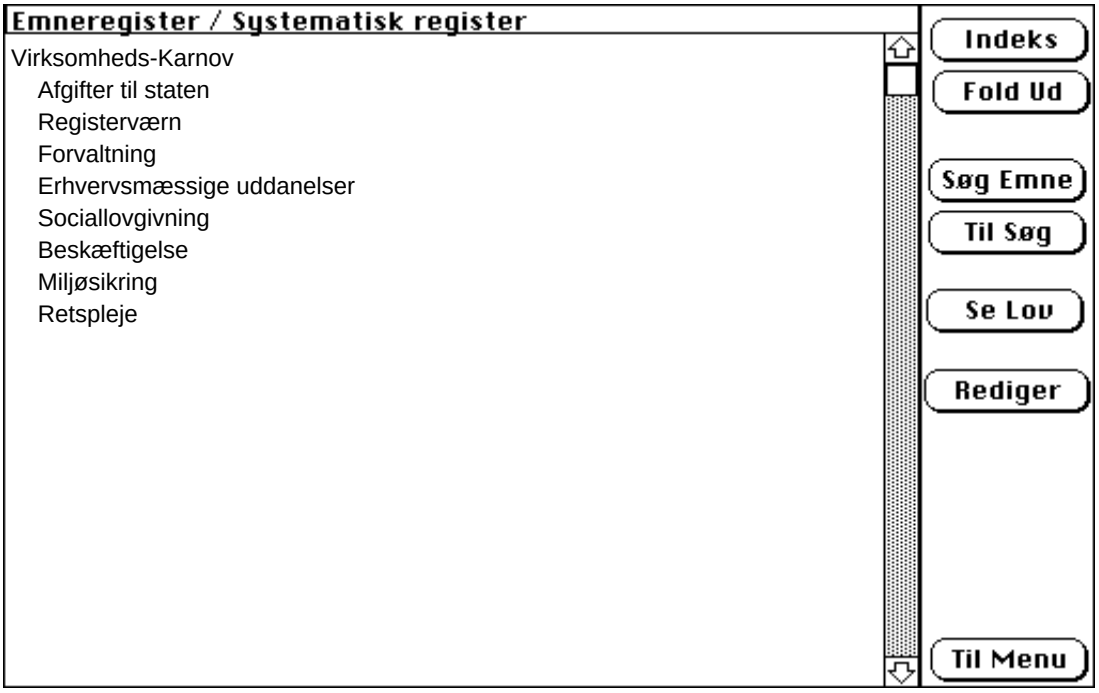
Figur 6.4. Resultatet af søgningen med søgeordet 'afgift%'. Søgningen omfatter alle love, og der søges i både lovtekster og noter.

I søge-skærmbilledet udskifter vi ordet 'moms%' med 'afgift%' for at se, om det giver bedre held. Med ordet 'afgift%' har vi gjort søgningen bredere, og vi udvider den yderligere ved tillige at søge i noterne. Dette markeres ved at klikke på Noter. Det er muligt, at der står noget i noterne, som kan føre os til den ønskede lovtekst.

Resultatet af denne søgning er 33 lovtekster og noter, hvor 'afgift%' forekommer. Oversigten over fundne forekomster synes Jens Peter Nielsen giver et godt overblik over søgningen. Svartiderne fandt alle deltagerne i demonstrationerne udmærkede til demonstrationsbrug; men i et færdigt system skal de være kortere.

Peter Ingwersen spørger, hvorfor oversigten ikke er ordnet sådan, at de nyeste tekster kommer først; det er det normale. Vi diskuterer forskellige muligheder for at ordne en sådan oversigt. I dette tilfælde vil det bedste sandsynligvis være at gruppere teksterne ud fra strukturen i lovgivningssystemet. Det vil betyde, at loven kommer først og derefter de tilhørende bekendtgørelser, cirkulærer og noter. Dette er i overensstemmelse med Per Sjøqvists krav til et søgesystem. Han mener, et søgesystem så vidt muligt skal følge og støtte den juridiske metode.

Det ses af oversigten, at 'afgift%' forekommer i mange lovtekster og noter. Det fremgår tydeligt, at oversigten indeholder irrelevante dokumenter, fx handler den første lov i oversigten om bortskaffelse af olie- og kemikalieaffald. Det er endvidere svært at overskue, hvilke dokumenter der muligvis er relevante. Problemet med de mange irrelevante dokumenter kan måske undgåes, hvis søgningens omfang afgrænses til en mindre del af lovteksterne. Søgning indenfor emner er mulig i Edb-Karnov. Vi klikker på Emne under Omfang for at markere, at der skal søges indenfor et emne. Derved sendes vi over i emneregistret, som kan bruges til at begrænse omfanget af en søgning.

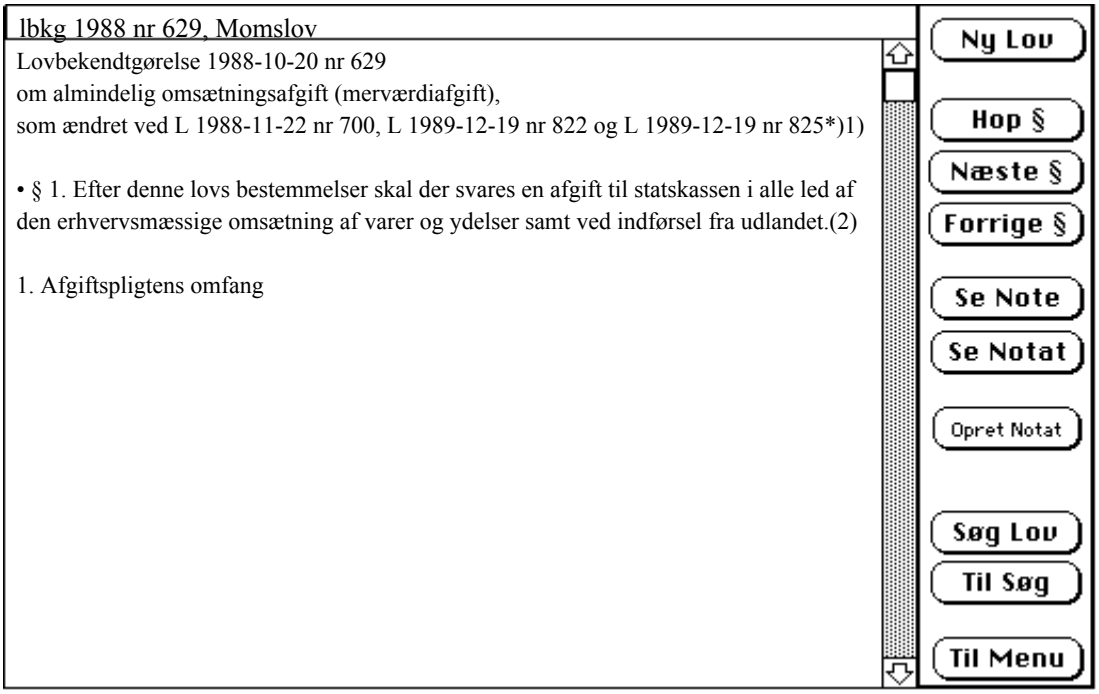


Figur 6.5. Figuren viser de otte emner på øverste niveau i emneregistret Virksomheds-Karnov.

I emneregistret er det første, vi gør, at markere Virksomheds-Karnov og klikke på Fold Ud. Herved foldes grundemneregistret Virksomheds-Karnov ud; det består, som det ses på figur 6.5, af otte emner. Da vi søger efter grundlaget for momsberegningen, er det første emne, Afgifter til staten, umiddelbart det bedst egnede. Afgifter til staten markeres, og der klikkes på Søg Emne. Herved afgrænses søgningen til det markerede emne, og vi kommer tilbage til søge-skærmbilledet.

Per Sjøqvist mener, at emneregistret kan være nyttigt til at fange de grå områder i lovgivningssystemet. De grå områder opstår, fordi mange lovtekster ikke kan placeres entydigt i det hierarki, lovgivningssystemet udgør. I Karnovs Lovsamling er emneregistret et hierarki, hvor hver lovtekst er placeret i ét retsområde. I Retsinformation opdeles lovteksterne i baser, således at hver lovtekst er placeret i ét ministeriums baser. I praksis er en lovtekst imidlertid ofte relevant i flere forskellige sammenhænge, fx både indenfor miljøområdet og indenfor landbrugsområdet. Edb-Karnovs emneregister giver mulighed for at afspejle, at en lovtekst er relevant indenfor flere områder.

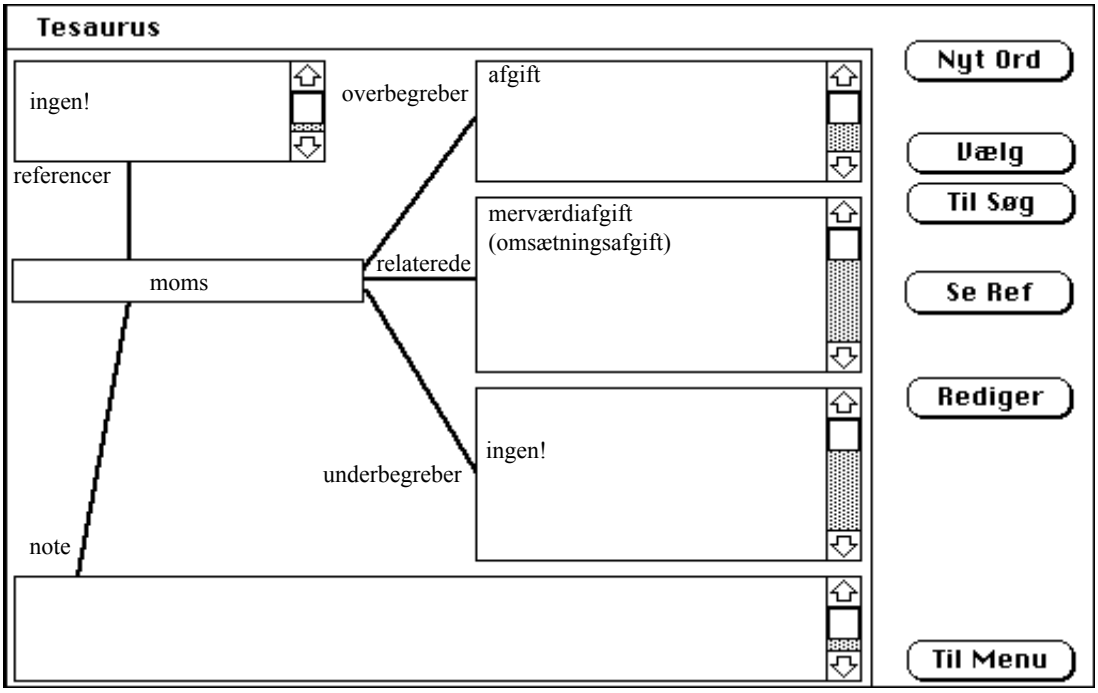
Af søge-skærmbilledet fremgår, at søgningens omfang nu er et emne, og overskriften fortæller, at dette emne er Afgifter til staten. For yderligere at indsnævre søgningen vil vi ikke søge efter 'afgift%' i noterne, men kun i lovteksterne. Søgningen resulterer i, at der findes 12 love og bekendtgørelser, hvoraf den ene er Momsloven. Ved at markerer Momsloven fås valget mellem at gentage søgningen indenfor loven for derved at finde paragraffer eller at se selve lovteksten. Vi vælger, at se selve lovteksten, og skifter derved til det skærmbilledet, hvor lovteksterne vises.



Figur 6.6. Lovtekst-skærmbilledet viser Momslovens hoved og dens første paragraf.

I Lovtekst-skærmbilledet fremvises starten på momsloven, hovedet og den første paragraf (se figur 6.6). Per Sjøqvist bemærker, at fremvisning af én paragraf er for lidt til at give det samme overblik som på papir. Det skal efter hans mening være muligt at se mindst én side.

Under gennemlæsningen af teksten på skærmen er det første, vi bemærker, at der i anden linie står, at loven handler om omsætningsafgift og merværdiafgift. Det tyder på, at moms omtales som omsætningsafgift eller merværdiafgift i lovteksterne. Omsætningsafgift og merværdiafgift er altså synonymer til moms. Det vil være hensigtsmæssigt for den fremtidige brug af Edb-Karnov, at det fremgår af tesaurussen, at moms, omsætningsafgift og merværdiafgift er synonymer. For at tilføje disse synonymer hopper vi til tesaurussen via Edb-Karnovs menuskærmbillede, dvs der klikkes på Til Menu og derefter på Tesaurus.

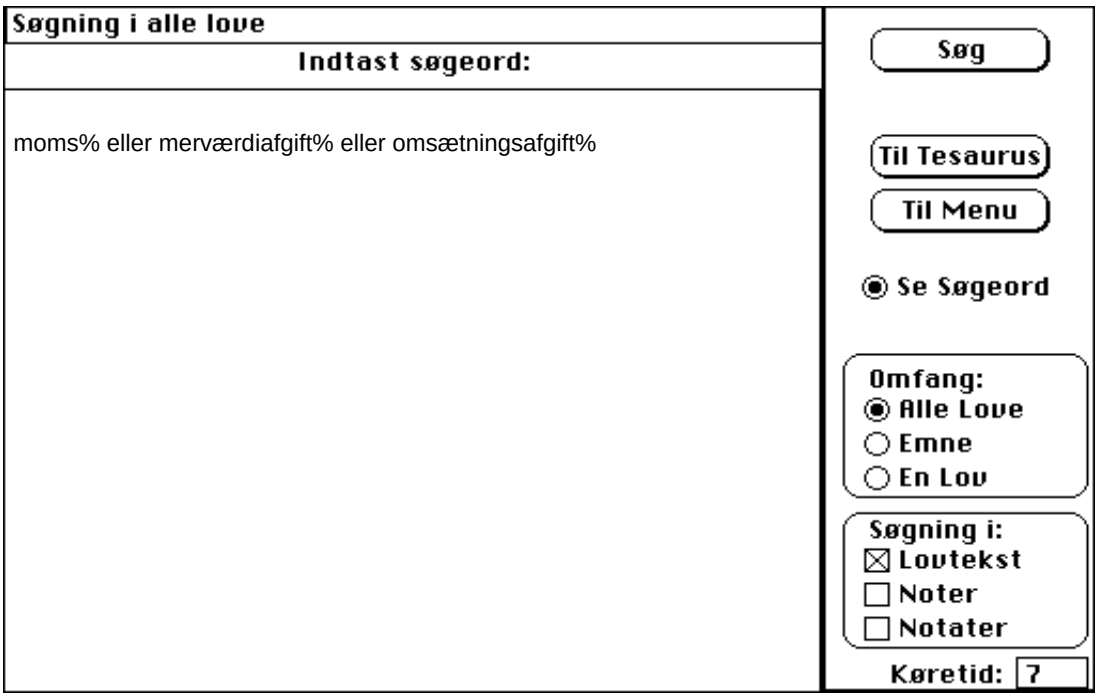


Figur 6.7. Skærmbilledet viser resultatet af tilføjelse af merværdiafgift og omsætningsafgift, som relaterede begreber til moms. Parentesen om omsætningsafgift angiver, at det er et forslag.

I tesaurussen ses overbegreber, underbegreber og relaterede begreber til moms, fra vi sidst var i tesaurussen. Feltet med relaterede begreber markeres, og der klikkes på Rediger. Derefter vælges Tilføj for at indsætte merværdiafgift og omsætningsafgift som relaterede begreber for moms. Vi indtaster merværdiafgift og klikker efterfølgende på Godkend, da vi ikke er i tvivl om, at moms og merværdiafgift er relaterede begreber. Ved indlæggelsen af omsætningsafgift klikker vi ikke på Godkend. Derved får dette relaterede begreb kun status som forslag. Begreber, der kun er indlagt som forslag, markeres på skærmen med en parentes. Resultatet af tilføjelsen af de to ord fremgår af figur 6.7. Per Sjøqvist synes, det er unødvendigt, at et ord først skal indlægges som forslag. Han tvivler på, at adskillelsen af forslag og tesaurussens etablerede begreber giver nogle væsentlige fordele.

For at illustrere værdien af løbende at kunne ændre i tesaurussen, vil vi nu vise, hvordan søgningen efter grundlaget for momsberegningen forløber med den ændrede tesaurus. Vi starter som før med den resultatløse søgning med det ene søgeord 'moms%'. Derefter slåes moms op i tesaurussen. De relaterede begreber kan overføres til forespørgslen ét ad gangen ved at markere ordet og klikke på Vælg eller samlet ved at klikke på Vælg og derefter markere, om det er overbegreber, underbegreber og/eller relaterede begreber, der skal overføres. Vi overfører dem ét ad gangen. Efter at have klikket på Vælg skal vi angive, om ordet skal overføres alene, med OG eller med ELLER. Da der er tale om synonymer, klikker vi på ELLER.

Per Sjøqvist mener, at brugeren skal udføre alt for mange funktioner for automatisk at overføre søgeord til forespørgslen. Han vil foretrække selv at indskrive søgeordene i forespørgslen. Peter Ingwersen kommer med et forslag, der vil forenkle overførslen af søgeord til forespørgslen. Relaterede begreber og underbegreber skal automatisk føjes til forespørgslen med ELLER - det giver ingen mening at bruge OG. Det er kun i forbindelse med overbegreber, valgmuligheden mellem at tilføje med OG eller med ELLER er relevant.

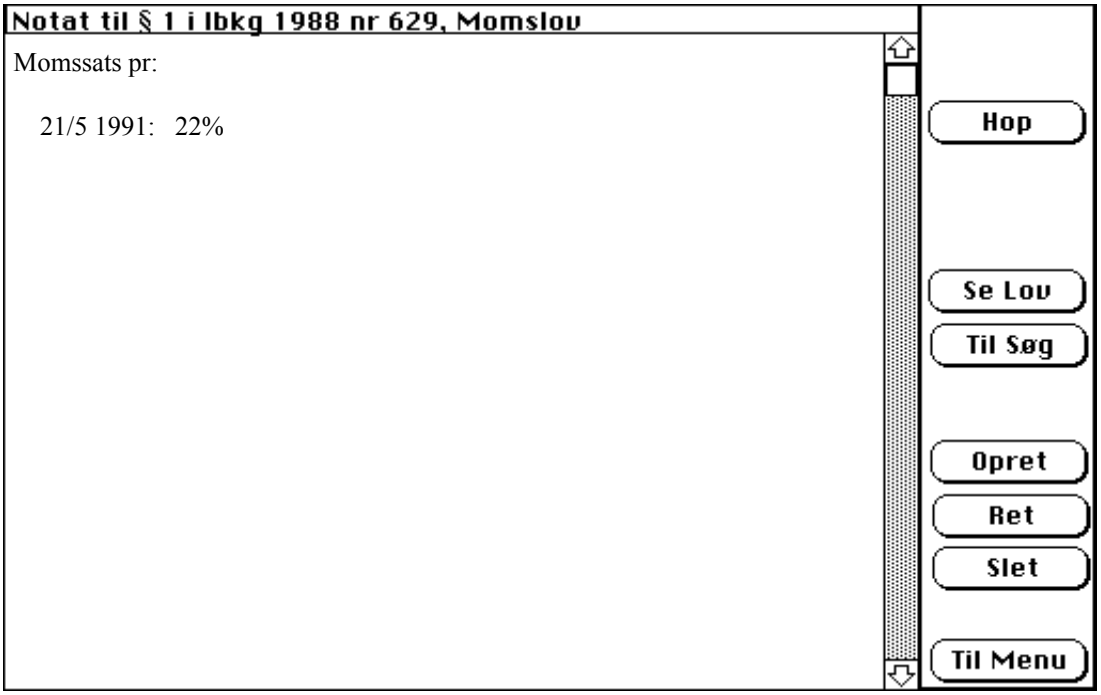


Figur 6.8. Forespørgsel hvor der søges på moms og dets relaterede begreber, merværdiafgift og omsætningsafgift, i alle love og udelukkende i lovtekster.

Tilbage i søgeskærmbilledet ses nu resultatet af overførslen af de to relaterede begreber for moms (se figur 6.8). Denne forespørgsel resulterer i 15 love og bekendtgørelser. Det ses umiddelbart, at Momsloven er en af dem, og dermed er grundlaget for momsberegningen lokaliseret. Momsloven markeres i oversigten, hvorved vi får valget mellem at gentage søgningen i denne lovtekst eller se den. Vi vælger at gentage søgningen og får derved en oversigt over, hvilke paragraffer der indeholder et af søgeordene.

Per Sjøqvist giver udtryk for, at ELLER er den væsentligste operator i angivelsen af forespørgslerne. Med OG-operatoren løber han en risiko for at frasortere et relevant dokument, fordi det kun indeholder det ene af de to søgeord. Han vil derfor foretrække, at forespørgslerne altid formuleres med ELLER, så han selv kan styre frasorteringen. Per Sjøqvist og Peter Ingwersen er enige om, at angivelsen af forespørgslerne med OG og ELLER ikke er hensigtsmæssig. De mener, OG og ELLER er mere relevant for bibliotekarer end for jurister, da bibliotekarer er specialister i søgning. Per Sjøqvist gør opmærksom på, at det kun er en meget lille del af juristers arbejde, der går med søgning i retskilderne. Det er kun 3-4 gange om ugen, han selv bruger registrene i Karnovs Lovsamling. Alle deltagerne i demonstrationerne synes, begrebsbaseret fremfinding (se afsnit 2.4) lyder som et godt alternativ til den måde, forespørgslerne skal angives på i Edb-Karnov.

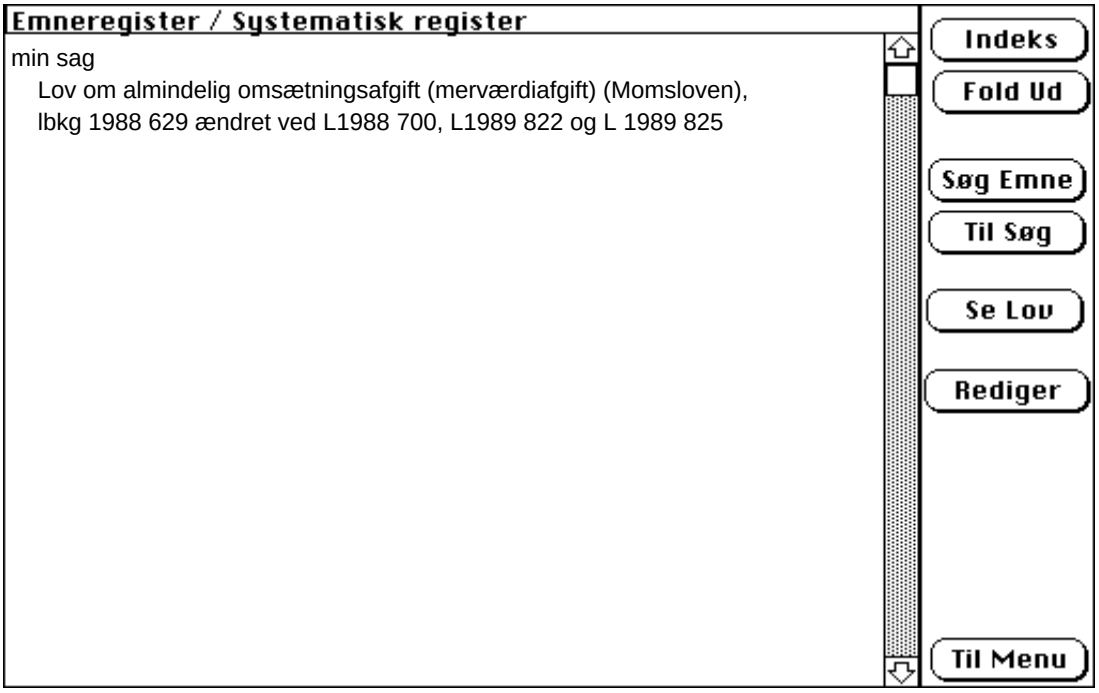
Jens Peter Nielsen forslår, at tesaurussen får en meget mere central placering i søgningerne. Der er for mange veje ind i Edb-Karnov: Søge-skærmbilledet, emneregistret og tesaurussen. Det er forvirrende; der burde kun være én vej ind i systemet. Efter Jens Peter Nielsens mening er det tesaurussen, forespørgslerne skal bygges op omkring. Han ser specielt en mulighed for integrere tesaurussen og emneregistret. Hvis der er knyttet henvisninger til et ord i tesaurussen, bør de ikke blot kunne bruges til opslag, men også til at begrænse søgningen til de pågældende dokumenter.



Figur 6.9. Der er oprettet et notat til momslovens første paragraf for at kunne beskrive ændringer i momssatsen. Notatet indeholder dog kun den i øjeblikket gældende momssats.

Efter at grundlaget for momsberegningen er lokaliseret, vil vi lave en oversigt over de forskellige momssatser, der har været gældende gennem tiden. I Edb-Karnov kan det gøres ved at oprette et notat til en af momslovens paragraffer. Vi hopper til momslovens første paragraf for at oprette et notat med momssatsen. I lovtekst-skærmbilledet klikkes på Opret Notat, hvorefter vi kommer til notat-skærmbilledet. Vi skriver, at momssatsen pr 21/5 1991 er 22% (se figur 6.9). Derefter klikkes på Opret, hvorefter notatet bliver lagt ind i databasen og gjort søgbart på fuldstændig samme måde som de øvrige tekster i Edb-Karnov.

Jens Peter Nielsen synes, notater er en virkelig god idé. Det skal dog være muligt at skelne mellem forskellige type af notater, fx personlige notater og virksomhedens notater. Virksomhedsnotater er blandt andet en nem måde at markere noget som værende relevant for andre end en selv. Datostempling af notaterne vil også være en fordel.



Figur 6.10. Figuren viser, at der er oprettet et emne, faktisk et nyt og selvstændigt emneregister, kaldet 'min sag', og at Momsloven er knyttet til dette emne.

Vi regner med, at den sag, eksemplet er begyndelsen på, bliver langvarig. Vi finder det derfor relevant at oprette et sagsorienteret emneregister, der skal dække denne sag. Efterhånden som sagen udvikler sig, vil de lovtekster, noter og notater, der bruges i forbindelse med sagen, blive føjet til dette emneregister. Vi opretter således et nyt emne, kaldet 'min sag', ved siden af grundemneregistret, og afslutter eksemplet med at knytte Momsloven til 'min sag' (se figur 6.10).

Per Sjøqvist mener først og fremmest, muligheden for at oprette egne sager er relevant sammen med notaterne. De to faciliteter kan bruges til at samle oplysninger, der har relation til givne, ofte forekommende, sager. Disse oplysninger kan fx være tidligere udarbejdede kontrakter eller oversigter over typiske erstatnings- og bødesatser. Per Sjøqvist ser gode muligheder i at bruge Edb-Karnov til en sådan vidensopsamling i forbindelse med typiske sager.

Efter at have gennemgået vores eksempel afsluttede vi demonstrationen med at vise, hvordan en paragraf kan eksporteres fra Edb-Karnov til et tekstbehandlingssystem. Det sker ved først at markere paragraffen i Edb-Karnov og kopiere den ud på Macintosh'ens indbyggede klippebord. Derefter hopper vi over i tekstbehandlingssystemet og henter paragraffen ind fra klippebordet. Per Sjøqvist mener, denne mulighed for at eksportere tekst kan være nyttig. Jens Peter Nielsen er synligt imponeret over, hvor let det er at importere og eksportere.

6.2 Sammenfatning

I dette kapitel har vi gennemgået et eksempel, der har illustreret Edb-Karnovs væsentligste funktioner. Eksemplet blev ligeledes brugt i vores demonstrationer af Edb-Karnov for nogle indbudte personer. Disse personers kommentarer er blevet inddraget i gennemgangen af eksemplet. Der er stor forskel på disse kommentarer; de spænder fra fundamental kritik af fuldtekstsøgning til bemærkninger i forbindelse med brugergrænsefladen. Vi har valgt at lade kritikken fremstå uden vores kommentarer, da

sådanne kommentarer hurtigt får karakter af et forsvar. Formålet med demonstrationerne af Edb-Karnov har netop været at få kritik, der kan bruges i den videre udvikling af Edb-Karnov eller et lignende system.

Det er vores vurdering, at deltagerne i demonstrationerne generelt mener, at Edb-Karnovs tesaurus, emneregister og notater er faciliteter, der støtter systemets udvikling over tid og vil være nyttige for fagfolk. Det er desuden vores fornemmelse, at deltagerne havde meget let ved at følge med i demonstrationen. Vi mener derfor, brugergrænsefladen har været letforståelig for dem. Peter Ingwersen fremhævede, at Edb-Karnov er en rigtig prototype, ikke blot en papirmodel eller en brugergrænseflade uden funktionalitet bag. Derved adskiller Edb-Karnov sig efter hans udsagn fra en stor del af de prototyper, der omtales i litteraturen.

Flere af deltagerne i demonstrationerne retter kritik mod fuldtekstsøgning; de foretrækker nøgleordsbaseret søgning i stedet. Vi er ikke enige i, at nøgleordsbaseret søgning er at foretrække for fuldtekstsøgning; men det har styrket vores formodning, om at et søge-system bør tilbyde både fuldtekstsøgning og nøgleordsbaseret søgning. Jens Peter Nielsen giver udtryk for, at han gerne ser tesaurussen placeret meget mere centralt i angivelsen af forespørgslerne. Vi finder dette forslag meget spændende, da vi i det ser vi en mulighed for at gøre tesaurussen til det fælles udgangspunkt for både fuldtekstsøgning og nøgleordsbaseret søgning. Vi tror, tesaurussen er velegnet til dette formål, da ordene i tesaurussen både kan støtte brugeren i valget af søgeord og fungere som nøgleord med tilknyttede henvisninger. Som Jens Peter Nielsen foreslår, kan nøgleordsfaciliteten i tesaurussen endvidere danne udgangspunkt for en sammensmeltning af tesaurussen og emneregistret. Vi kunne fx forestille os, at brugeren forud for at søge fuldtekst med forespørgslen 'moms%' havde mulighed for at afgrænse søgningen til de dokumenter, der henvises til fra 'afgift' - overbegrebet for moms.

Per Sjøqvists pointering, af at søgning kun er en meget lille del af juristers arbejde, støtter vores formodning om, at det er afgørende, at Edb-Karnov er integreret med de andre edb-systemer, der indgår i den juridiske sagsbehandling. Vi har behandlet integrationen af Edb-Karnov og tekstbehandling, og vi har nævnt muligheden for at etablere referencer mellem Edb-Karnov og et journaliseringssystem. Per Sjøqvists pointering peger imidlertid også på, at en grundliggende empirisk undersøgelse af juristers arbejde vil være relevant. Formålet med en sådan undersøgelse skulle være at finde ud af, hvad jurister faktisk bruger deres tid på, for derved bedre at blive i stand til at udvikle hjælpemidler, der støtter dem i deres arbejdssituation.

Endelig diskuterede vi brugen af relationsdatabaser kontra søgesystemer baseret på inverterede filer med Peter Ingwersen. Han mener, det er hurtigere og nemmere at udvikle søgesystemer med inverterede systemer, mens udvidelser og vedligeholdelse er lettere, hvis søgesystemet er baseret på en relationsdatabase. Denne diskussion har styrket vores tro på, at en relationsdatabase er et velegnet grundlag for et fuldtekstsøgesystem som Edb-Karnov.

7. Arbejdsmetode

I dette kapitel vil vi redegøre for den arbejdsmetode, der ligger til grund for specialet. Vi vil resumere de metodeovervejelser, vi har gjort i løbet af de tidligere kapitler. Derudover vil vi prøve at evaluere arbejdsmetoden, dels for at vurdere det grundlag specialet er baseret på, dels for at opsamle eventuelle erfaringer til senere brug. Dette kapitel er sammenfatningen på vores fremgangsmåde og metodeovervejelser under udarbejdelsen af specialet. I næste kapitel vil vi sammenfatte de resultater, vi har nået.

Specialets opbygning afspejler vores arbejdsproces. Vores vejleder Erik Frøkjær havde nogle ideer til, hvordan fuldtekstsøgesystemer baseret på relationsdatabaser kunne støtte fagfolk i deres informationssøgning. Vi har realiseret og videreudviklet disse ideer gennem en case om juridisk informationssøgning. For at fastholde de ideer, der var udgangspunktet for specialet, valgte vi at sætte os ind i de tekniske aspekter ved fuldtekstsøgesystemer, før vi påbegyndte den case, vi ville bruge til at udvikle og evaluere ideerne. På grund af vores vejleders kendskab til juridisk sagsbehandling kunne vi vælge denne rækkefølge uden at risikere, at arbejdet med de tekniske aspekter ville føre os langt væk fra den brugssituation, vores case handlede om. Vi mener, den valgte rækkefølge har været udbytterig for os. Vi har hele tiden været opmærksomme på helheden; men ved at vente med at indkredse juridisk sagsbehandling, fik vi tid til at sætte os ind i mulighederne med fuldtekstsøgning uden at lade os styre af den konkrete anvendelse.

Vi havde ikke tidligere beskæftiget os med informationssøgesystemer. Vores kendskab til juridisk sagbehandling var ligeledes meget begrænset. Vi skulle derfor til at sætte os ind i nogle nye områder. I dette kapitel behandler vi først de overvejelser, der ligger til grund for litteraturstudiet og derefter vores indkredsning af juridisk sagsbehandling. Efter det tages de værktøjer, prototypen er udviklet med, op til en kort vurdering. Kapitlet afsluttes med en sammenfatning.

7.1 Litteraturstudie

Da vi begyndte på specialet, havde vi intet kendskab til informations- eller fuldtekstsøgning. Vi kendte ikke en grundbog, der kunne føre os ind i emnet, og havde heller ikke hørt om nogen, som arbejdede indenfor feltet. Vi startede derfor litteraturstudiet helt fra bunden. For at opnå et både bredt og grundigt kendskab til fuldtekstsøgning, planlagde vi et omfattende og systematisk litteraturstudie. Under litteraturstudiet har vi:

- Gennemsøgt ACM's årlige litteraturguide for artikler, bøger og *proceedings* under emnet 'H.3 Information Storage and Retrieval'. På dette punkt omfatter vores litteraturstudie årene 1985-89.
- Gennemset alle numre af to centrale tidskrifter, *Information Processing & Management* og *ACM SIGIR Forum* (SIGIR står for 'Special Interest Group on Information Retrieval'), for årene 1985-89.
- Gennemset to årlige publikationer for årene 1981-89. Det drejer sig om konferencerapporten fra den årlige hovedkonference om forskningen i informationssøgning, "Research and Development in Information Retrieval", og "Annual Review of Information Science and Technology" (ARIST).

Udvælgelsen af artikler, bøger og *proceedings* fra ACM's årlige guide skete alene ud fra deres titel. De udvalgte titler var ikke altid dækkende for indholdet; nogle viste sig at være

irrelevante. Omkring halvdelen af de ca 160 titler, vi havde udvalgt i første omgang, blev således kasseret igen. Udvælgelse af artikler alene ud fra titlen betyder endvidere, at vi kan have overset relevant litteratur, fordi vi ikke fandt titlen interessant og relevant for vores problemstilling.

Vi har forsøgt at sikre fremfinding af den væsentligste litteratur såvel for årene 1985-89 som før, ved under læsningen at være opmærksomme på litteraturreferencer. I den sammenhæng har ARIST, der består af oversigtsartikler, været en væsentlig støtte. Hvis vi skulle starte et litteraturstudie i dag, ville vi tage udgangspunkt i disse oversigtsartikler.

Information Processing & Management - det ene af de tidskrifter vi gennemså hvert nummer af for årene 1985-89 - indeholdt mange gode artikler. Dette tidsskrift er specielt rettet mod forskningen i informationssøgning, og mange af forskningsresultaterne publiceres i dette tidsskrift. *ACM SIGIR Forum* - det andet tidskrift vi gennemså hvert nummer af for årene 1985-89 - har vi ikke benyttet meget. Det indeholdt stort set kun artikler, der var irrelevante for vores speciale.

Det har været en lærerig proces at gennemsøge et forskningsområde for at finde den litteratur, der er relevant for netop vores problemstilling. Vores litteratursøgning har betydet, at vi har fundet artikler, som vi sikkert aldrig ville have set, hvis udgangspunktet havde været en grundbog. Vi mener, vi har fået en bredere indsigt og større forståelse af området ved vores fremgangsmåde.

Den litteratur, vi har fundet, læst og benyttet, belyser hovedsageligt de tekniske aspekter ved informationssøgesystemer. Det er kun en lille del af vores litteratur, der behandler brugssituationen. Det skyldes, at vi valgte primært at rette vores litteraturstudie mod de tekniske aspekter. Der er skrevet store mængder litteratur om baggrunden for, at mennesker retter forespørgsler til søgesystemer, om brugerens søgeadfærd osv. Den side af forskningen i informationssøgning dækkes ikke af vores litteraturstudie.

Et andet karakteristisk træk ved specialets litteratur er, at den i vid udstrækning handler om nøgleordsbaseret søgning. Det skyldes primært, at fuldtekstsøgning er et nyt område, som først nu er ved at tage form (Tenopir & Ro 1990), mens nøgleordsbaseret søgning har været et forskningsområde siden sidst i 50'erne. En stor del af de begreber og søgefaciliteter, fuldtekstsøgning er baseret på, stammer således fra nøgleordsbaseret søgning. De to områder har mange fælles problemstillinger, og de fleste af dem er blevet indgående behandlet, mens nøgleordsbaseret søgning endnu var enerådende. Forskellen på de to typer informationssøgning kan ud fra et teknisk synspunkt siges primært at bestå i, at indholdet af søgeordstabellen er forskelligt. En stor del af den litteratur, vi har fundet om fuldtekstsøgning, har været evalueringer af eksisterende kommercielle systemer. Disse evalueringer kan være spændende og inspirerende; men der er mange spørgsmål og problemer, de overhovedet ikke kommer ind på. De har derfor kun været af begrænset interesse for os. Vi har dog fundet nogle gode artikler om fuldtekstsøgning; vi vil specielt nævne (Blair & Maron 1985) og (Tenopir 1984). I litteraturen om nøgleordsbaseret søgning har vi imidlertid fundet en lang række artikler, der er relevante i forbindelse med fuldtekstsøgning. Sammenfattende mener vi, specialet er baseret på et solidt litteraturgrundlag.

7.2 Juridisk sagsbehandling

Vi valgte - som tidligere nævnt - at foretage litteraturstudiet om fuldtekstsøgning, før vi udbyggede vores kendskab til juridisk sagsbehandling. Da vi begyndte specialet, havde vi kun et meget overfladisk kendskab til juridisk sagsbehandling. Dette kendskab måtte uddybes for at vi kunne udforme og gennemføre vores case. Vi har opnået vores kendskab til juridisk sagsbehandling gennem samtaler med vores vejleder og Jens Peter Nielsen,

gennem en demonstration af Retsinformation, gennem litteratur, og ikke mindst gennem interview med to jurister. Vi har ikke lavet et systematisk litteraturstudie om juridisk sagsbehandling; vores litteratur om dette emne er da også ret begrænset. Det skyldes, at det primære formål med vores case var at udvikle og evaluere vores ideer om udvikling over tid, ikke at undersøge selve brugssituationen.

Inden vi foretog de to interview og dermed fik et væsentligt bedre indblik i juridisk sagsbehandling, var vi ind i mellem frustrerede over vores overfladiske kendskab til juridisk sagsbehandling. Vi frygtede, at vi var ved at bevæge os langt væk fra den brugssituation, vi stræbte efter at støtte. I disse tilfælde støttede vi os til, at vores ideer havde en generalitet, der rakte ud over vores case. Det var imidlertid først og fremmest på grund af vores vejleder og vores samtaler med Jens Peter Nielsen tidligt i forløbet, at vi undgik at fjerne os fra den brugssituation, vores case udspillede sig i.

For at opnå et større kendskab til juridisk sagsbehandling foretog vi to kvalitative interview. Vores krav til interviewpersonerne var, at de skulle være jurister og bruge lovtekster i deres arbejde. Derimod måtte de ikke have med udfærdigelsen af lovtekster at gøre. De to personer, vi interviewede, var Dorthe la Cour og Per Sjøqvist. Dorthe la Cour er jurist i Dansk Arbejdersgiverforening, hvor hun rådgiver foreningens medlemsvirksomheder. Per Sjøqvist er advokat og medindehaver af advokatfirmaet Horten & Co. Begge interviewene varede ca 1 time og var løst strukturerede. De forløb som styrede samtaler, uden at der var nogle spørgsmål som absolut skulle besvares. Forberedelsen af interviewene skete ved udarbejdelse af en interviewguide (se bilag 3). Interviewguiden blev ikke fulgt slavisk, men snarere brugt som støtte og rettesnor. Vi optog de to interview på bånd. Den senere behandling af interviewene bestod i, at vi skrev et lettere bearbejdet og tematiseret referat af dem. Begge referater er gennemlæst og godkendt af interviewpersonerne. De kom med mindre rettelser så som fejl og misforståelser; men intet er bortcensuret. De godkendte referater findes i bilag 4.

Det kvalitative interview giver mulighed for at få en nuanceret fremstilling af nogle problemstillinger, som de tager sig ud for interviewpersonen. Det betyder, at nye og uventede problemstillinger kan dukke op. En begrænsning ved kvalitative interview er, at det er svært at afgøre, hvor repræsentative interviewpersonerne er. I den forbindelse vil vi nævne et par punkter, der skal tages med i vurderingen af de udtalelser, vores to interviewpersoner er kommet med.

Dorthe la Cour er positiv overfor edb-teknologi og ser nogle muligheder for, at den kan støtte hende i hendes arbejde. Per Sjøqvist er meget mere skeptisk overfor edb-teknologien. Den skal bevise sin berettigelse i forhold til de eksisterende hælpemidler i form af bøger og lignende. Det er vores fornemmelse, at Per Sjøqvist er mere repræsentativ end Dorthe la Cour med hensyn til juristers syn på edb-teknologi. Vi baserer blandt andet denne fornemmelse på Remmen (1989).

Det skal også bemærkes, at de to interviewede jurister er beskæftiget med forskellige typer sagsbehandling: Dorthe la Cour rådgiver, mens Per Sjøqvist fører sager. Det er ikke klart for os, hvor forskellige disse to typer sagsbehandling er, og i hvilket omfang de er repræsentative for juridisk sagsbehandling om helhed.

Vores indkredsning af juridisk sagsbehandling har givet os et vist kendskab til den brugssituation, vores case udspiller sig i. Vi mener, vores kendskab til juridisk sagsbehandling er tilstrækkeligt til, at Edb-Karnov er udviklet på et fornuftigt grundlag. Vi vil karakterisere vores kendskab som en fornemmelse af juridisk sagsbehandling. Mange af de faciliteter, der mangler før Edb-Karnov er et færdigt system, kræver efter vores mening et grundigere kendskab til juridisk sagsbehandling, fx kræver en forfining af brugergrænsefladen et nøjere kendskab til den juridiske metode og arbejdsgangen i den juridiske sagsbehandling. I det videre arbejde med udviklingen af fuldtekstsøgesystemer

til jurister vil en empirisk undersøgelse af juridisk sagsbehandling således være frugtbar.

7.3 Værktøjer

I valget af værktøjer var det afgørende for os, at det blev så let som muligt at lave en prototype, som kunne bruges til at udvikle og evaluere vores ideer om faciliteter, der tillader søgesystemer at udvikle sig over tid. Prototypen skulle for det første stimulere og konkretisere udviklingen af vores ideer. For det andet skulle prototypen nå et omfang og en funktionalitet, som var rimeligt i demonstrationssammenhæng. Det er således ikke afgørende for os, om fx svartiderne kunne være blevet mindre med et andet valg af værktøjer. Derudover er det tidskrævende at undersøge relevante værktøjer og svært at gøre det ordentligt på det tidspunkt, hvor valget skal træffes - mens ideerne endnu ikke er gennemarbejdede, og kravene til værktøjerne derfor er uklare. Vi valgte værktøjer uden at gå ind i en undersøgelse af, hvilke alternativer vi havde. Udviklingen af og arbejdet med Edb-Karnov har ikke givet os anledning til at fortryde denne fremgangsmåde.

De valgte værktøjer har ikke givet større problemer i udviklingen af protypen. Oracle og C har fungeret godt, specielt har grænsefladen mellem dem ikke givet problemer. De problemer, vi har haft, har først og fremmest været med Hypercard. De to væsentligste problemer er, at kortstørrelsen er fast, og at der kun kan være ét åbent vindue ad gangen. Derudover har vi også savnet en datastruktur som tabeller. Hypercard har imidlertid været utrolig velegnet til hurtigt og let at udvikle en grafisk brugergrænseflade med en omfattende funktionalitet. Vi mener, Hypercard har været overordentlig velvalgt til vores opgave. I denne vurdering spiller det naturligvis ind, at det ikke er brugergrænsefladen, vi har fokuseret på. Jensen (1990), der har lavet en prototype på en hypertekstudgave af Karnovs Lovsamling, valgte X-windows og UNIX-miljø for at få den ønskede funktionalitet og fleksibilitet i brugergrænsefladen. Vi er helt enige i, at Hypercard er utilstrækkeligt i forbindelse med et færdigt system og vil blot bemærke, at såvel X-windows som UNIX kan fås til Macintosh.

Værktøjerne kan også vurderes på, hvor lang tid udviklingen af prototypen har taget, og hvor meget programkode den har givet anledning til. Vi har brugt ca 3 1/2 måned på at udvikle prototypen, inklusiv den måned vi brugte på at sætte os ind i værktøjerne. Resultatet er ca 40 sider C-programkode og ca 200 sider Hypercard-programkode (på grund af programkodens omfang har vi ikke vedlagt den som bilag). Sammenholdt med Edb-Karnovs funktionalitet mener vi, dette viser, at værktøjerne har været udmærkede til formålet.

I implementeringen af Edb-Karnov har vi naturligvis benyttet de manualer, der hører til de enkelte værktøjer. Derudover har vi haft glæde af Kerninghan & Ritchie's bog om C (Kerninghan & Ritchie 1988) og Goodman's bog om Hypercard (Goodman 1987). Manualerne til Oracle har været tilstrækkelige til vores brug; men hvis det bliver aktuelt med yderligere optimering af databasen, er de mangelfulde.

7.4 Sammenfatning

I dette kapitel har vi sammenfattet og beskrevet de væsentligste elementer i den arbejdsmetode, der ligger til grund for specialet. Intentionen med arbejdsmetoden har været at støtte vores bestræbelser på at udvikle og evaluere vores ideer om søgesystemer, der kan tilpasses de løbende ændringer i brugerens arbejdssitiuation. For at fastholde disse ideer valgte vi at undersøge de tekniske aspekter ved konstruktionen af fuldtekstsøgesystemer, før vi vendte os mod den brugssituation, vores case udspillede sig i. Denne beslutning har været vellykket.

Vi mener, opdelingen af specialet i en beskrivelse af *state of the art* for fuldtekstsøgesystemer - baseret på litteraturstudiet - og en case har givet plads til vores ideer. Vi indledte vores case med en indkredsningen af juridisk sagsbehandling. Denne indkredsning består først og fremmest af to interview. Derefter udviklede vi en prototype på et juridisk fuldtekstsøgesystem. Prototypen er udviklet ved hjælp af Oracle, C og Hypercard. Disse tre værktøjer har været udmærkede til vores formål; men i forbindelse med et færdigt system er Hypercard utilstrækkeligt. Et væsentligt formål med prototypen var, at den skulle nå et omfang og en funktionalitet, som gjorde det muligt at bruge den i demonstrationssammenhæng. Vi har fået meget ud af at demonstrere Edb-Karnov for andre og diskutere vores ideer med dem. Sammenfattende vurderer vi, at den valgte arbejdsmetode har været frugtbar og i overensstemmelse med vores intentioner med specialet.

8. Sammenfatning

Dette speciale har handlet om fuldtekstsøgesystemer, der skal fungere som redskaber for fagfolk. Et sådant søgesystem skal efter vores mening være et personligt redskab og dermed søges integreret i brugerens arbejdssituation. Vi har primært fokuseret på ét basalt forhold i arbejdssituationen: Den ændrer sig over tid. Hvis det skal lykkes at integrere et søgesystem i brugerens arbejdssituation, er det efter vores mening nødvendigt, at disse ændringer kan afspejles i søgesystemet. Vi ser således et stort behov for faciliteter, der giver mulighed for, at disse ændringer indarbejdes i søgesystemet, så snart brugeren bliver opmærksom på dem.

Specialet består af en beskrivelse af *state of the art* for fuldtekstsøgesystemer og en case indenfor området juridisk informationssøgning. I beskrivelsen af *state of the art* var det de tekniske aspekter ved fuldtekstsøgesystemer, vi lagde vægt på. Brugssituationen blev inddraget, da vi vendte os mod vores case. Vi indledte vores case med at forsøge at indkredse juridisk sagsbehandling og de krav, ønsker og forbehold, jurister møder juridiske informationssøgesystemer med. Derefter designede og implementerede vi en prototype på et juridisk fuldtekstsøgesystem, Edb-Karnov, med henblik på at analysere og vurdere vores ideer om faciliteter, der tillader søgesystemer at udvikle sig over tid. Resultatet er tre sådanne faciliteter - et dynamisk emneregister, en dynamisk tesaurus og muligheden for at tilføje egne notater. Derudover har vi undersøgt realismen i at basere søgesystemer, der støtter udvikling over tid, på relationsdatabaser. I forlængelse af prototypen har vi diskuteret - men ikke implementeret - et forslag til, hvordan de ændringer, brugeren foretager løbende, kan integreres med udgivelsen af nye udgaver af Edb-Karnov. Endelig har vi demonstreret Edb-Karnov for såvel læseren som en potentiel udbyder, en potentiel bruger og en forsker indenfor informationssøgning.

I dette kapitel vil vi først resumere Edb-Karnovs faciliteter gennem en sammenligning med de faciliteter, der typisk findes i andre fuldtekstsøgesystemer. Derefter vender vi os mod de to centrale ideer i specialet. Vi sammenfatter først de resultater, vi har nået med hensyn til faciliteter, der støtter søgesystemers udvikling over tid. Umiddelbart derefter tager vi realismen i at basere søgesystemer på relationsdatabaser op til vurdering. Til slut vender vi os mod en problemstilling, som er opstået og vokset i løbet af specialet: Behovet for at integrere fuldtekstsøgning og nøgleordsbaseret søgning.

8.1 Fuldtekstsøgesystemer og Edb-Karnov

Fuldtekstsøgesystemer baseres næsten altid på inverterede filer. Ved inverteringen oprettes en søgeordstabel bestående af alle de ikke-stopord, der indgår i dokumenterne. Det er denne søgeordstabel, der er grundlaget for behandlingen af forespørgslerne. Forespørgslerne er brugernes væsentligste mulighed for at angive, hvad de søger efter. Men derudover er det ofte muligt at afgrænse søgningerne til en del af søgesystemets dokumenter. Denne mulighed udspringer af, at dokumenterne ofte er organiseret i en række baser, således at hvert dokument findes i netop én base. Hvis dokumenterne er opdelt i baser, kan søgesystemet give brugerne nogle mere eller mindre avancerede muligheder for at afgrænse søgningerne til én eller flere udvalgte baser.

Den søgeordstabel, der oprettes ved inverteringen, indeholder enkeltord. Det betyder, at der umiddelbart kun kan søges på enkeltord, ikke på vendinger. I de fleste tilfælde er det dog muligt at søge på vendinger ved hjælp af nærhedsoperatoren NABO. Boolsk søgning kombineret med nærhedsoperatorer er den mest udbredte søgeteknik i forbindelse

med fuldtekstsøgning. Derudover giver fuldtekstsøgesystemer næsten altid mulighed for at bruge joker-operatorer i formuleringen af forespørgslerne. Hensigten med joker-operatorer er at reducere forespørgslernes følsomhed overfor variationer i søgeordenes bøjningsformer, stavemåde osv.

Edb-Karnov er et fuldtekstsøgesystem og har derfor en række lighedspunkter med andre fuldtekstsøgesystemer; men på flere punkter adskiller Edb-Karnov sig væsentligt fra andre fuldtekstsøgesystemer. Til lighedspunkterne hører, at søgeteknikken i Edb-Karnov er boolsk søgning kombineret med nærhedsoperatorer. Efter vores mening er boolsk søgning den mest anvendelige af de tre muligheder, vi har behandlet - boolsk søgning, udvidet boolsk søgning og skimming. Vi har implementeret boolsk søgning sådan, at forespørgslerne angives som en følge af søgeord adskilt af boolske operatorer. Under demonstrationerne af Edb-Karnov har flere peget på, at denne brugergrænseflade ikke kan bruges i et endeligt system. Vi er helt enige i denne kritik og har peget på begrebsbaseret fremfinding som en mulighed for at give boolsk søgning en væsentligt bedre brugergrænseflade. Ved begrebsbaseret fremfinding formuleres forespørgslerne som en fællesmængde af begreber, hvor hvert begreb beskrives ved hjælp af en række termer. Når vi har valgt den simple brugergrænseflade, er det blandt andet fordi, vi har ønsket at kombinere boolsk søgning med nærhedsoperatorer.

Efter vores mening afhænger nærhedsoperatorers anvendelighed af, om de refererer til en kontekst med en vis selvstændig betydning. Det skyldes, at kontekster med en vis selvstændig betydning i højere grad kan forventes at indeholde alle tekstens centrale begreber. NABO refererer til vendinger, og vendinger har en vis selvstændig betydning. Derudover bør valget af nærhedsoperatorer afhænge af det specifikke område, søgesystemet er rettet mod. I Edb-Karnov har brugeren nærhedsoperatoren 'indenfor samme paragraf' til rådighed udover de boolske operatorer OG og ELLER.

I Edb-Karnov er det også muligt at begrænse søgningerne til en del af systemets dokumenter. Grundlaget for denne facilitet er emneregistret. Emneregistret er en selvstændig facilitet og således uafhængigt af, hvordan dokumenterne er organiseret i baser eller lignende. En konsekvens af det er, at emneregistret er et netværk fremfor et hierarki. I et netværk kan det samme dokument optræde i flere af de grupper, emneregistret fastlægger. Derved undgåes mange af de noget tilfældige valg af, hvilken gruppe et dokument skal placeres i. Indenfor lovgivningen er dét fx relevant, hvis et dokument vedrører både landbrugsområdet og miljøområdet.

Et centralt problem ved fuldtekstsøgning er, at brugeren er afhængig af ordvalget i dokumenterne. Problemet skyldes, at søgesystemerne kun behandler data. Et dokument findes således kun frem, hvis der er et eksakt match mellem søgeordene og ordene i dokumentet. Brugeren betragter imidlertid ikke søgeordene som data, men som meningsbærende begreber. Ex: For brugeren er 'moms', 'omsætningsafgift' og 'merværdiafgift' typisk det samme, fordi disse tre ord har næsten samme betydning; for et søgesystem er de altid forskellige, fordi de staves forskelligt. Et tilsvarende problem findes i forbindelse med et ords forskellige bøjningsformer. For at reducere dette problem - springet fra begreb til term - giver Edb-Karnov brugeren mulighed for at bruge en tesaurus og joker-operatorer i formuleringen af forespørgslerne. Med tesaurussen kan brugeren få støtte i valget af søgeord og udvide sine forespørgsler med fx et søgeords underbegreber. Efter vores mening er en on-line tesaurus nødvendig for, at fuldtekstsøgning kan fungere tilfredsstillende. Tesaurussen er imidlertid en af de faciliteter, der adskiller Edb-Karnov fra flertallet af fuldtekstsøgesystemer.

Edb-Karnov adskiller sig også fra andre fuldtekstsøgesystemer ved at give mulighed for tilføjelse af egne notater. Med notaterne kan kommentarer, henvisninger, nye tekster osv føjes til de tekster, systemet allerede indeholder. Herved bliver det muligt at samle en

større del af de tekster, brugeren har behov for at søge i, på ét sted. Med muligheden for at tilføje egne notater skifter Edb-Karnov karakter fra at være et søgesystem adskilt fra det øvrige arbejde henimod at være integreret i juristens samlede arbejdssituation. Vi har implementeret egne notater som én generel type tekster; i et fuldt udbygget system bør der skelnes mellem flere forskellige typer. Vi har diskuteret bogmærker som eksempel på en speciel type notater.

Mange af forskellene på Edb-Karnov og andre fuldtekstsøgesystemer udspringer af, at Edb-Karnov er et søgesystem til fagfolk. Det betyder, at der er mange ad-hoc forespørgsler og kun få forespørgsler, der gentages rutinemæssigt hver måned eller lignende. Fagfolk er endvidere ikke indstillet på at ofre selve søgesystemet ret megen opmærksomhed. Det stiller store krav ikke blot til brugergrænsefladen, men også til de underliggende funktioner og systemets fleksibilitet. Vores fokusering på udvikling over tid er et forsøg på at skille vedligeholdelsen af søgesystemet i to: En del, der skal varetages af udbyderen, og en del, brugeren med fordel kan gøre selv. Fagfolk har kompetencen til selv at forestå en væsentlig del af den løbende vedligeholdelse, der skal sørge for, at søgesystemet glider så ubemærket som muligt ind i arbejdssituationen. Vi har gennem hele specialet søgt at udvikle og evaluere faciliteter, der giver den enkelte bruger mulighed for selv at varetage systemets udvikling over tid. Disse faciliteter samt realismen i at basere fuldtekstsøgesystemer på relationsdatabaser behandles i de to følgende afsnit.

Endelig er vi i løbet af specialet blevet klar over, at et søgesystem bedst tilfredsstiller brugerens behov ved både at omfatte fuldtekstsøgning og nøgleordsbaseret søgning. Det er fuldtekstsøgning, der er emnet for dette speciale; men Edb-Karnov giver også muligheden for en enkel form for nøgleordsbaseret søgning: Opslag på henvisninger knyttet til ordene i tesaurussen. I afsnit 8.4 vil vi se lidt nærmere på, hvordan fuldtekstsøgning og nøgleordsbaseret søgning kan integreres.

8.2 Udvikling over tid

Vi har gennem hele specialet undersøgt mulighederne for at give et fuldtekstsøgesystem en funktionalitet og fleksibilitet, der sætter den enkelte bruger i stand til at varetage systemets udvikling over tid. Central styring og fastlæggelse af ajourføringerne er efter vores mening ikke tilstrækkeligt; de enkelte brugere skal have mulighed for løbende at foretage netop de ændringer, de finder relevante. Det er der to grunde til: For det første er de ajourføringer, udbyderen foretager, forsinkede i forhold til ændringerne i brugssituationen. For det andet udgiver udbyderen et generelt produkt; brugeren vil således ofte have glæde af at kunne udbygge og nuancere systemet indenfor sit specielle område.

Muligheden for at foretage ændringer efter eget behov er det mest udtalte eksempel på, at Edb-Karnov er et søgesystem til fagfolk. Fagfolk har kompetencen til at bruge og administrere en sådan dynamik. Brugere uden faglig kompetence løber en væsentligt større risiko for at foretage ændringer, der nok passer til den aktuelle situation, men ikke har nogen generel relevans eller gyldighed.

Udvikling over tid er et perspektiv, vi kun har set taget op et enkelt sted i litteraturen - i forbindelse med vedligeholdelsen af en tesaurus (Güntzer m.fl.1989). Vi har derfor søgt at lokalisere de steder i et søgesystem, hvor udvikling over tid med fordel kan inddrages. Derefter har vi forsøgt at konkretisere vores forestillinger ved at udvikle og implementere faciliteter, der tillader et søgesystem at udvikle sig over tid. Vi har lokaliseret og udviklet tre faciliteter: Et dynamisk emneregister, en dynamisk tesaurus og egne notater. I relation til udvikling over tid er det centrale ved disse tre faciliteter de tilknyttede redigeringsfunktioner.

I emneregistret giver redigeringsfunktionerne brugeren mulighed for at ændre i den systematik, teksterne er opdelt efter. Specielt kan brugeren anvende redigeringsfunktionerne til at opbygge sine egne sagsorienterede emneregistre. I tesaurussen er redigeringsfunktionerne baseret på, at der arbejdes med forslag. Da Edb-Karnovs facilitet til nøgleordsbaseret søgning er placeret i tesaurussen, er det muligt at tilføje nye nøgleord og henvisninger. Muligheden for at tilføje egne notater åbner Edb-Karnov mod den øvrige del af brugerens arbejdssituation. Her giver redigeringsfunktionerne mulighed for at knytte en uddybende kommentar til indholdet af en paragraf, for at henvise til en relevant sag i et journaliseringssystem osv.

De tre faciliteter, der gør det muligt for brugeren løbende at tilpasse systemet til ændringer i arbejdssituationen, har været lette at implementere. Det er således i høj grad realistisk at inddrage udvikling over tid i et søgesystem, når det baseres på en relationsdatabase. Vores omend meget begrænsede demonstrationer tyder endvidere på, at kompetente fagfolk finder disse faciliteter nyttige. Faciliteterne er endnu ikke udviklet til deres endelige form; specielt brugergrænsefladen - som vi har afgrænset os fra at gå i dybden med - skal der arbejdes mere med. De tre faciliteter har imidlertid nået en form, der efter vores mening viser, at ideerne er frugtbare. Vi konkluderer, at et godt søgesystem skal omfatte faciliteter, der støtter udvikling over tid.

Behovet for, at den enkelte bruger kan foretage ændringer, betyder imidlertid ikke, at centrale ajourføringer kan undværes. Emneregistret, tesaurussen og egne notater vil ikke blive brugt, hvis ændringerne går tabt, når en ny udgave af Edb-Karnov installeres. Et afgørende aspekt i forbindelse med et søgesystems udvikling over tid er derfor integrationen af de løbende ændringer og udgivelsen af nye udgaver.

Vi er kommet med et forslag til, hvordan denne integration kan foretages i forbindelse med fx grundtesaurussen. Essensen i dette forslag er at betragte tesaurussen som en grundtesaurus med nogle filtre ovenpå. Filtrene indeholder alle de ændringer, brugeren foretager i tesaurussen. Ved at adskille brugerens ændringer fra grundtesaurussen bliver det for det første muligt at genskabe grundtesaurussen. For det andet bliver det muligt at installere en ny udgave af grundtesaurussen uden, at brugerens egne ændringer går tabt. Hvis brugeren ikke har benyttet redigeringsfunktionerne, foregår installationen automatisk. Ellers må brugeren involveres i de tilfælde, hvor der er konflikt mellem de ændringer, brugeren selv har foretaget, og de ændringer, der introduceres med den nye udgave. Det skal bemærkes, hvilket niveau disse konflikter forekommer på: Der er udelukkende tale om konflikter mellem knuder og/eller kanter i den graf, der repræsenterer tesaurussen. Uoverensstemmelser mellem meningen med brugerens ændringer og ajourføringen mener vi ikke kan behandles maskinelt.

8.3 Relationsdatabaser som grundlag for fuldtekstsøgesystemer

Mulighederne for at foretage ændringer - i emneregistret, i tesaurussen og i form af egne notater - kombineret med et krav om effektive søgninger stiller store krav til de værktøjer, systemet baseres på. Vi mener, brugen af en relationsdatabase som grundlag for Edb-Karnov har været en succes. Fordelene i form af funktionalitet og fleksibilitet mere end opvejer de ulemper, relationsdatabaser kritiseres for at have. Det har været let at implementere såvel Edb-Karnovs statiske faciliteter som de dynamiske. Det er ved de dynamiske faciliteter, vi har gjort mest brug af relationsdatabasens særegne funktionalitet og fleksibilitet. Brugen af en relationsdatabase har således været et godt grundlag for at udvikle et søgesystem med en funktionalitet, der sigter på at støtte udvikling over tid. Edb-Karnov er udviklet i trin - først de grundlæggende funktioner, så emneregistret, derefter tesaurussen og til sidst egne notater. Det skal bemærkes, at databasen er på såvel

tredje som fjerde normalform, og at de udvidelser, der er sket på hvert trin, har været meget lette at føje til det, vi allerede havde implementeret. Specielt er tilføjelsen af egne notater et eksempel på, at det er let at tilføje en ny type tekster. Vi mener således, at brugen af en relationsdatabase både har bidraget til at gøre implementeringen af Edb-Karnovs funktioner enkel og vil bidrage til at gøre systemet lettere at udvide og vedligeholde.

Relationsdatabaser kritiseres for at kræve for meget tid og plads til at kunne konkurrere med søgesystemer baseret på inverterede filer. Hovedårsagen til disse to problemer er, at relationsdatabaser af hensyn til normalformerne opdeles i et stort antal tabeller. Problemet med tidsforbruget skyldes således primært, at de *joins*, der efterfølgende må bruges for at kombinere data fra forskellige tabeller, er mange og tidskrævende. Problemet med pladsforbruget skyldes, at de nøgler, der sammenkæder oplysningerne i de forskellige tabeller, lagres i alle de tabeller, de indgår i. Det store pladsforbrug skyldes også, at dataene ofte lagres både i tabellerne og i de indeks, det er nødvendigt at oprette af hensyn til svartiderne.

Vi må give kritikerne ret i, at pladsforbruget er stort. De godt 400 sider tekst, der er lagt ind i Edb-Karnov, udgør ca 4 Mb i en flad fil. I Edb-Karnov er disse 400 sider lagret i såvel teksttabellerne som søgeordstabellerne og deres indeks, ialt knap 19 Mb. Brugen af en relationsdatabase fører til et ekstra pladsforbrug på næsten 400% i forhold til det, selve teksten fylder. Til sammenligning er det ekstra pladsforbrug 50-300% ved brug af inverterede filer. Vi mener imidlertid ikke, et stort pladsforbrug er noget problem, da prisen på selv meget store mængder lagerplads er relativt lille og til stadighed falder. Hvis Edb-Karnov skulle udvides til at omfatte alle de ca 4000 sider i Karnovs Lovsamling, ville en 300 Mb harddisk stadig være rigeligt, og sådan en koster trods alt kun 22.800 kr excl moms (maj 1991).

Mens pladsforbruget er stort, behøver svartiderne efter vores mening ikke blive et problem. Edb-Karnov er en prototype og udviklet med henblik på at fungere til demonstrationsbrug. De krav, vi har sat til svartiderne, er bestemt af, at de skal være rimelige i den sammenhæng. Vi har opnået svartider på ca 30 sekunder for de mest komplicerede søgninger, og det har kun krævet meget beskedne optimeringer at komme dertil. Der er således mange muligheder for at optimere svartiderne yderligere. Ex: I øjeblikket bygges alle SQL-forespørgsler i Edb-Karnov op om samme grundforespørgsel. Det betyder i flere tilfælde, at der genereres unødigt komplicerede SQL-forespørgsler. Der kunne i stedet udvikles en specifik SQL-forespørgsel for hver type søgning - en når der søges med nærhedsoperatoren 'indenfor samme lovtekst', en anden når søgningen afgrænses til et retsområde osv. Det vil give mulighed for at tune hver enkelt SQL-forespørgsel til netop den type søgninger, den skal behandle.

Med baggrund i vores prototype mener vi, det er både realistisk og fordelagtigt at basere en edb-udgave af hele Karnovs Lovsamling på en relationsdatabase. Vi har ikke grundlag for at udtale os om, hvorvidt relationsdatabaser med fordel kan anvendes til søgesystemer, der omfatter meget store tekstmængder. De ca 4000 tætskrevne sider i Karnovs Lovsamling har imidlertid en berettigelse i sig selv. Vi mener derfor, kritikken af relationsdatabasers brug i forbindelse med søgesystemer er for unuanceret. Det er muligt, at ulemperne ved relationsdatabaser er udtalte i søgesystemer med meget store tekstmængder; men det er i hvert fald nødvendigt at inddrage tekstmængdens størrelse i diskussionen. Relationsdatabaser er efter vores mening velegnede som grundlag for søgesystemer med Edb-Karnovs omfang og funktionalitet.

8.4 Begrebsbaseret søgning

Udgangspunktet for specialet var, at vi ville behandle problemer med og ideer til udvikling af fuldtekstsøgesystemer med fokus på faciliteter, der tillader systemerne at

udvikle sig over tid. Efterhånden, som vi har arbejdet med sådanne faciliteter og fået konkretiseret vores ideer i en prototype, er behovet for søgesystemer, der både omfatter fuldtekstsøgning og nøgleordsbaseret søgning blevet stadig tydeligere for os. Fremfor at stille fuldtekstsøgning og nøgleordsbaseret søgning op overfor hinanden - sådan som mange gør, og vi selv gjorde i starten - tror vi, det vil være frugtbart at rette blikket mod, hvad vi kunne kalde begrebsbaseret søgning. Kernen i begrebsbaseret søgning er en integration af fuldtekstsøgning og nøgleordsbaseret søgning. I det følgende vil vi sammenfatte vores argumenter for, at en sådan integration er frugtbar, og skitsere det første skridt i retning af begrebsbaseret søgning.

Nøgleordsbaseret søgning bygger efter vores mening på en antagelse om, at dokumentsamlingen er velstruktureret. For at kunne bestemme en præcis og dækkende mængde nøgleord for hvert dokument skal hvert af dem behandle få og velafgrænsede emner, og hvert emne må kun behandles i få dokumenter. Hvis det er tilfældet, er nøgleordsbaseret søgning velegnet til søgning efter få centrale dokumenter - i disse tilfælde er der overensstemmelse mellem den måde, nøgleordene er tildelt på, og formålet med søgningerne. En del undersøgelser viser, at nøgleordsbaseret søgning giver højere *precision* end fuldtekstsøgning, så ved søgning efter få centrale dokumenter er nøgleordsbaseret søgning at foretrække fremfor fuldtekstsøgning.

Fuldtekstsøgning forudsætter ikke nogen struktur og forsøger heller ikke at etablere nogen. Ved fuldtekstsøgning foregår søgningerne direkte i dokumenterne; der lægges ingen systematik ind mellem brugeren og dokumenterne. Fuldtekstsøgning er derfor velegnet i forbindelse med søgninger, der ikke følger dokumenternes hovedlinier, men går på tværs eller skal lokalisere detaljer. I litteraturen er der endvidere nogenlunde enighed om, at fuldtekstsøgning giver et højere *recall* end nøgleordsbaseret søgning.

Fuldtekstsøgning og nøgleordsbaseret søgning supplerer således hinanden. Et godt søgesystem bør derfor stille begge muligheder til rådighed. Yderligere et argument for dette er, at springet fra begreb til term, der er et centralt problem i forbindelse med fuldtekstsøgning, i vid udstrækning undgåes ved nøgleordsbaseret søgning. I Edb-Karnov har vi kun suppleret fuldtekstsøgningen med en enkel form for nøgleordsbaseret søgning. Det skyldes, at det overordnede emne for specialet er fuldtekstsøgning. I tesaurussen ser vi imidlertid en mulighed for at komme et stykke i retning af en integration af fuldtekstsøgning og nøgleordsbaseret søgning. Vi tror, tesaurussen er velegnet til dette formål, da ordene i tesaurussen både kan støtte valget af søgeord og fungere som nøgleord med henvisninger til Edb-Karnovs dokumenter. Det er afgørende for integrationen af fuldtekstsøgning og nøgleordsbaseret søgning, at de to typer søgning udgør forskellige muligheder på én fælles vej ind i systemet. Vi ser en mulighed for at gøre tesaurussen til det fælles udgangspunkt for alle søgninger.

Den centrale idé i dette speciale er udvikling over tid. Vi har søgt at udvikle og evaluere faciliteter, der giver brugeren mulighed for at tilpasse et søgesystem til de løbende ændringer i brugssituationen. Udvikling over tid er ikke et perspektiv, der ofres ret meget opmærksomhed i litteraturen. Vi finder det afgørende for et søgesystems anvendelighed, at det integreres i brugerens arbejdssituation. I denne integration mener vi, udvikling over tid er et centralt aspekt. Dette understreges yderligere af, at mange edb-systemer har en lang levetid. På baggrund af vores prototype konkluderer vi, at det er både realistisk og frugtbart at inddrage udvikling over tid i et søgesystems funktionalitet.

Litteraturliste

Ashford, John H. (1986):
"Integrating text and non-text: Why is it important?" i (Kimberley 1986), p3-20.
Baggrunden for artiklen er, at fuldtekstsøgesystemer typisk kun omfatter tekst - figurer, billeder og lignende er ikke med. Denne adskillelse af tekst og ikke-tekst er kunstig og skyldes dels begrænsninger i teknologien, dels at kun få har prøvet at integrere tekst og ikke-tekst. Artiklen behandler behovet og mulighederne for en sådan integration.

Ashford, John H. (1987):
"Text storage and retrieval in the ORACLE relational database management system: design study and intended applications" i *Program*, vol.21 nr.2 (april 1987), p108-123.
Artiklen udspringer af et udviklingsprojekt som British Telecom New Information Services og Oracle Corporation UK har startet. Formålet med projektet er at tilføje faciliteter til Oracle, der gør det bedre egnet til at behandle tekst. Artiklen handler om, hvad der adskiller tekst fra strukturerede data, og hvilke specielle operationer, der er behov for i forbindelse med behandling af tekst.

Bache, Christian (1991):
"Ny teknik i rettens tjeneste" i *Juristen* (udgives af DJØF) vol.73 nr.3, p93-106.
I artiklen beskrives, hvor edb bruges indenfor retssektoren. Anvendelserne omfatter fx edb i tinglysningen, tekstbehandling, edb som sagsbehandlingsredskab i fogedretten, edb til retskildesøgningen og såkaldte ekspertsystemer. Om edb i retskildesøgningen nævnes specielt, at søgesystemerne bør støtte både den brede, grundlægggende søgning ud fra et givet tema og den mere rutinemæssige kontrol af den seneste udvikling på et iøvrigt kendt område.

Bancilhon, Francois & DeWitt, David J. (red.) (1988):
"Very Large Data Bases, VLDB. Proceedings of the fourteenth international conference on very large data bases" (Los Angeles, California 1988), Morgan Kaufman Publishers Inc., Palo Alto, California. 490 sider. ISBN 0-934613-75-3.

Batty, David (1989):
"Thesaurus construction and maintenance: A survival kit" i *Database* vol.12 nr.1 (februar 1989), p13-20.
Artiklen handler om, hvordan tesaurusser kan konstrueres og vedligeholdes, hovedvægten ligger på konstruktionen. Batty lægger vægt på, at det overvejes, hvem brugerne er, og hvad de har behov for; det har betydning for hvilken bredde og dybde, tesaurussen bør have. Først derefter kan selve konstruktionen begynde. Der skal vælges ord til tesaurussen, og de skal arrangeres i grupper og undergrupper. Til sidst fastlægges de eksakte relationer mellem ordene. Batty nævner blandt andet relationen HN ('Historical Note'), der angiver ændringer i et ords brug eller form.

Belkin, Nicholas J. (1980):
"Anomalous state of knowledge as a basis for information retrieval" i *Canadian Journal of Information Science* vol.5, p133-143.
Artiklen beskriver i tæt overensstemmelse med (Mackay 1960), hvad der er baggrunden for, at mennesker stiller forespørgsler til søgesystemer. Denne baggrund defineres som en tilstand med utilstrækkelig viden: *The anomalous state of knowledge* (ofte forkortet ASK).

Belkin, Nicholas J. & Croft, W. Bruce (1987):
"Retrieval Techniques", p109-145 i Martha E. Williams (red.): "Annual Review of Information Science and Technology" (ARIST) vol.22, American Society for

Information Science (ASIS).
En glimrende oversigt over og kort gennemgang af de mest betydningsfulde søgeteknikker, såvel eksakt match som partielt match. Artiklen tager udgangspunkt i en klassifikation af søgeteknikker. I konklusionen fremhæves, at det forekommer utilstrækkeligt at behandle alle forespørgsler med én søgeteknik; fremtidige systemer bør baseres på integration af flere søgeteknikker.

Belkin, Nicholas J. & van Rijsbergen, C. J. (red.) (1989):
"Research and Development in Information Retrieval. Proceedings of the twelfth annual international ACM SIGIR conference (Cambridge, MA, 23-25 june, 1989)", ACM Press, New York. 257 sider. ISBN 0-89791-321-3.
Proceedings fra 1989-udgaven af den årlige hovedkonference om forskningen i informationssøgning.

Belkin, Nicholas J. & Vickery, Alina (1985):
"Interaction in Information Systems", Library and Information Research Report no.35 (LIRR 35), The British Library, Boston Spa, Wetherby, West Yorkshire. 250 sider. ISBN 0 7123 3050 X.
Vi har kun benyttet bogens andet kapitel. Det handler om det informationsbehov, der går forud for og giver anledning til informationssøgningen. Her beskrives på grundlag af et meget omfattende litteraturstudie, hvordan forskernes syn på dette informationsbehov har ændret sig siden 50'erne. Belkin & Vickery beskriver forskernes syn på informationsbehovet ved hjælp af en tredeling, som stort set svarer til de tre paradigmer i (Ingwersen 1987).

Biller, Horst (1983):
"On the Architecture of a System Integrating Data Base Management and Information Retrieval" i (Salton & Schneider 1983), p80-97.
Biller diskuterer mulighedenrne for at integrere DBMS, specielt relationsdatabasesystemer, og IRS. Han konkluderer, at der kræves flere ændringer i relationsdatabasesystemerne, før effektive DBMIRS - som det integrerede produkt kaldes - kan konstrueres. Men han ser også en udvikling i retning af, at disse ændringer foretages.

Bing, Jon (1981):
"Text retrieval in Norway" i *Program*, vol.15 nr.3 (juli 1981), p150-162.
Artiklen giver en oversigt over den norske forskning i informationssøgning med særlig vægt på juridisk informationssøgning. Blandt de emner, der diskuteres, er juristers informationsbehov i relation til juridiske søgesystemer og begrebsbaseret fremfinding. Ved begrebsbaseret fremfinding formuleres forespørgslerne som en fællesmængde af forskellige ideer. Hver idé beskrives i forespørgslen ved en række begreber. Et konkret resultat af denne tilgang er, at søgesystemet blot beder om de begreber, der beskriver hver ide, derefter genererer det selv den boolske forespørgsel med OG, ELLER og de nødvendige parenteser.

Bing, Jon (1984):
"Rettslige informatjonssystemer", p15-27 i Cecilia Magnusson & Olav Torvund (red.): "Juristen i datasamhället. Nordisk årsbok for rättsinformatik 1984", P. A. Norstedt & Söners Forlag, Stockholm.
Artiklen er en introduktion til edb-baseret juridisk informationssøgning. De edb-baserede søgesystemer sammenlignes med de velkendte søgesystemer, dvs bøger og tidsskrifter. Sidst i artiklen gives en oversigt over de edb-baserede juridiske søgesystemer i Norden: Rättsdata i Sverige, Finlex i Finland, DC-jura og DATA LEX i Danmark og Lovdata i Norge.

Blair, David C. (1986):
"Full Text Retrieval: Evaluation amd Implications" i *International Classification* vol.13 nr.1, p18-23.
Beskriver Blair & Maron's evaluering af fuldtekstsøgesystemer. I denne artikel fremlægges noget materiale, som ikke er med i (Blair & Maron 1985), først og fremmest: De relevante dokumenter opdeles af brugerne i vitale, tilfredsstillende og marginale. *Recall* beregnes for hver af disse grupper; men under den antagelse, at databasen indeholder mindst ét vitalt dokument for hver forespørgsel, giver

det ingen ændringer af undersøgelsens resultater. Se iøvrigt (Blair & Maron 1985).

Blair, David C. (1988):
"An extended relational document retrieval model" i *Information Processing & Management* vol.24 nr.3, p349-371.
En introducerende artikel. Blair er positiv overfor relationsdatabasesystemernes funktionalitet, fx autorisationskontrol og rapportgeneratorer, og vurderer, at SQL er godt til konstruktion af ad-hoc forespørgsler. Plads- og tidsforbrug diskuteres kort, og det bemærkes, at det interessante spørgsmål er, om RDBMS er tilstrækkeligt effektive, ikke om RDBMS er mere eller mindre effektive end hierarkiske databaser og netværksdatabaser.

Blair, David C. & Maron, M. E. (1985):
"An evaluation of retrieval effectiveness for a full-text document-retrieval system" i *Communications of the ACM* vol.28 nr.3 (marts 1985), p289-299.
Blair & Maron's eksperimenter er blandt de få, der er foretaget på store databaser. Hovedresultatet er, at brugerne selv troede, de opnåede et *recall* på 75%, mens det faktisk kun var ca. 20%. Til gengæld var *precision* omkring 79%. Den eneste forklaring, Blair & Maron kan finde, er, at det lave recall må skyldes generelle svagheder ved fuldtekstsøgning.

Blume, Peter (1989):
"Juridisk informationssøgning. En praktisk introduktion til retssystemets kilder", 3. udgave, Akademisk Forlag, Odense. 207 sider. ISBN 87-500-2797-2.
Blume giver en grundig og introducerende beskrivelse af retskilderne og relationerne mellem dem. Grundloven indtager en særstilling 'over' alle andre retskilder. I juristers arbejde er de centrale retskilder imidlertid love og bekendtgørelser. Udover disse spiller også cirkulærer, tidligere domsafgørelser, betænkninger og lignende en rolle. Mulighederne for edb-baseret informationssøgning behandles i et særskilt kapitel.

Blume, Peter (1991):
"Offentliggørelsens betydning for retskildeværdi" i *Juristen* (udgives af DJØF) vol.73 nr.1, p12-20.
Artiklen handler om, hvilken betydning retskildernes offentliggørelse i egnede tidsskrifter, elektroniske informationssøgesystemer og lignende kan tillægges. Som eksempel er offentliggørelse ikke nogen gyldighedsbetingelse for cirkulærer. Det skyldes, at cirkulærer i princippet kun er bindende for myndighederne; i praksis er det imidlertid ofte i cirkulærerne, at *"den egentlige normering af retsforholdene finder sted"*. Blume vurderer, at udbygningen af Retsinformation vil øge betydningen af de offentliggjorte retskilder.

Bookstein, Abraham (1983):
"Explanation and generalization of vector models in information retrieval" i (Salton & Schneider 1983), p118-132.
Bookstein indleder med en sammenligning af boolsk søgning og den partielle søgeteknik term-vægtning. Derefter forklares term-vægtning og generaliseres til også at omfatte de probabilistiske søgeteknikker.

Brooks, H. M. (1987):
"Expert systems and intelligent information retrieval" i *Information Processing & Management* vol.23 nr.4, p367-382.
Brooks diskuterer, hvad vi skal forstå ved intelligent informationssøgning: Det indebærer, at søgesystemet benytter repræsenteret viden om sin verden af dokumenter, emner, brugere osv til at udlede, hvilke dokumenter der er relevante for hver enkelt brugers informationsbehov. Brooks konkluderer, at resultaterne af IR-forskernes interesse for AI og specielt ekspertsystemer endnu primært er ændrede holdninger og fokuseringer.

Buchan, R. L. (1989):

"Intertwining thesauri and dictionaries" i *Information Services & Use* vol.9 nr.3 (Oktober 1989), p171-175.
Buchan trækker formålet med tesaurusser og ordbøger/leksika op og argumenterer for, at der er fordele ved at integrere de to.

Burke, James J. & Ryan, Bob (1989):
"Gigabytes On-line" i *Byte*, vol.14 nr.10 (oktober 1989), p259-264.
Artiklen er en af fire i det pågældende nummer af *Byte* om de optiske teknologier. I artiklen beskrives udviklingen indenfor de optiske lagringsmedier - fra CD-ROM over WORM ('Write Once Read Many times') mod de sletbare optiske diske. Derudover beskrives teknologien bag de tre typer optiske diske overfladisk.

Bush, Vannevar (1945):
"As we may think" i *Atlantic Monthly* vol.176 nr.1 (juli 1945), p101-108. Artiklen er senere genoptrykt 5 gange:
1) På p16-38 i "Endless Horizons", Public Affairs Press, Washington D.C. (1946). 2) På p19-41 i H. S. Sharp (red.): "Readings in Information Retrieval", Scarecrow Press, New York (1964). 3) På p23-35 i M. Kochen (red.): "The Growth of Knowledge: Readings on Organization and Retrieval of Information", Wiley, New York (1967). 4) På p47-59 i Z. W. Pylyshyn (red.): "Perspectives on the Computer Revolution", Prentice-Hall, Englewood Cliffs, NJ (1970). 5) På p17-34 i Irene Greif (red.): "Computer Supported Cooperative Work: A book of readings", Morgan Kaufmann Publishers, San Mateo, Californien (1988). 783 sider. ISBN 0-934613-57-5.
En meget omtalt artikel, der især blandt amerikanske forskere betragtes som starten på moderne informationsvidenskab. Bush lancerer ideen om Memex ('memory extender'), et personligt fuldtekstsøgesystem, hvor brugeren kan lagre alle sine papirer, noter, artikler osv. Se også (Smith 1981).

Chiaramella, Y. (red.) (1988):
"Research & Development in Information Retrieval. 11th international conference (Grenoble, France, 13-15 June 1988)", ACM, New York, 677 sider. ISBN 2-7061-0309-4.
Proceedings fra 1988-udgaven af den årlige hovedkonference om forskningen i informationssøgning.

Codd, Edgar F. (1970):
"A Relational Model of Data for Large Shared Data Banks" i *Communications of the ACM* vol.13 nr.6 (juni 1970), p377-387.
Codd's oprindelige artikel, hvor ideen om at basere databaser på relationer præsenteres for første gang. Codd behandler logisk og fysisk datauafhængighed, han giver en generel specifikation af et forespørgselssprog til relationsdatabaser, og han kommer ind på redundans og konsistens. Artiklen blev ved udgivelsen review'et af Elliott (1971).

Codd, Edgar F. (1982):
"Relational Database: A Practical Foundation for Productivity" i *Communications of the ACM* vol.25 nr.2 (februar 1982), p109-117.
Artiklen er den tale, Codd holdt i anledning af, at han i 1981 fik ACM's Turing Award. Codd sammenfatter indholdet og styrkerne ved relationsdatabaser. Blandt styrkerne fremhæves datauafhængigheden, *views* og muligheden for at stille ad hoc-forespørgsler. Endelig bemærker Codd, at relationsdatabaser endnu er bedst egnet til strukturerede data.

Crawford, Robert G. (1981):
"The Relational Model in Information Retrieval" i *Journal of the American Society for Information Science* (ASIS) vol.32 nr.1, p51-64.
En tidlig introduktion til relationsdatabaser indenfor IR. Crawford gør rede for de fire første

normalformer og giver mange eksempler på IR-forespørgsler formuleret i såvel SQL som andre relationelle sprog. Crawford anerkender relationsdatabasernes velkendte fordele, men pointerer, at det er besværligt at formulere selv enkle IR-forespørgsler i SQL. Besværlighederne skyldes primært, at transformationen til 4NF fører til et stort antal relationer.

Croft, W. Bruce (1987):
"Approaches to intelligent information retrieval" i *Information Processing & Management* vol.23 nr.4, p249-254.
Artiklen er indledningen til et specialnummer om intelligent informationssøgning. Croft pointerer, at IR-forskerne har nået nogle resultater, som fortjener bedre end at blive glemt til fordel for AI-forskningens resultater. Derudover lokaliserer Croft inspirationen fra AI til tre områder: Ekspertsystemer, vidensrepræsentation og automatisk behandling af naturligt sprog.

Cross, George R. & DeBessonet, Cary G. (1985):
"Representation of legal knowledge for conceptual retrieval" i *Information Processing & Management* vol.21 nr.1, p35-44.
Artiklen beskriver søgesystemer til juridisk informationssøgning. Den starter med en kort beskrivelse af de traditionelle systemer - JURIS, LEXIS og WESTLAW. Derefter behandles forskellige forskningsprojekter, der arbejder med vidensrepræsentation indenfor det juridiske område. Cross & DeBessonet har meget store forhåbninger til, hvad AI-forskningen kommer til at betyde for juridisk informationssøgning.

Crouch, Carolyn J. (1988):
"A cluster-based approach to thesaurus construction" i (Chiaramella 1988), p309-320.
Crouch undersøger muligheden for at konstruere tesaurusser automatisk ved hjælp af klyngeteknikker og principper, der er udviklet i forbindelse med automatisk indeksering. Hun konkluderer, at resultaterne er lovende; men vi finder de overvejelser, der går forud for implementeringen, mest interessante.

Date, C. J. (1986):
"An Introduction to Database Systems" vol.1, 4. udgave, Addison-Wesley, Reading, Massachusetts, 639 sider. ISBN 0-201-19215-2.
En meget udbredt grundbog om databasesystemer.

"Dictionary of Computing" 3. udgave, Oxford University Press, New York (1990). 510 sider. ISBN 0 19 853825 1.
Et udmærket opslagsværk med anerkendte definitioner af mange centrale begreber.

Dreyfus, Hubert L. & Dreyfus, Stuart E. (1986):
"Mind over Machine. The Power of Human Intuition and Expertise in the Era of the Computer", Basil Blackwell, Oxford. 231 sider. ISBN 0-631-15126-5.
Bogen er et direkte angreb på AI-forskningens grundlag, idet den argumenterer overbevisende for, at det er principielt umuligt at formalisere eksperters kvalifikationer. Grundlaget for Dreyfus & Dreyfus's argumenter er en 5-trinsopdeling af menneskers tilegnelse af færdigheder: Novicen, der benytter kontekstfri regler; den viderekomne, der inddrager situationen; den kompetente, der tilføjer et formål; den kyndige, der forstår intuitivt, men verificerer med regler; og eksperten, der ikke længere arbejder analytisk, men helt intuitivt.

"Edb-ordbog" 1. udgave, dansk standard DS 2049-1970 fra Dansk Standardiseringsråd (DS), Gjellerup (1971). 154 sider. ISBN 87 13 01220 7.
Et udmærket opslagsværk med danske definitioner af mange centrale begreber. Ordbogen indeholder også ordlister med oversættelser fra engelske til danske ord, og omvendt.

Eddison, Betty & Batty, David (1988):

"Words, words, words - descriptors, subject headings, index terms" i *Database*, vol.11 nr.6 (december 1988), p109-113.

Artiklen beskriver en række af de problemer ved fuldtekstsøgning, som en tesaurus sigter på at reducere. I forbindelse med de kommercielt tilgængelige fuldtekstsøgesystemer konstateres, at udbyderne med fuldtekst er sluppet for at bruge tid og penge på at indeksere teksterne. I stedet er brugerne blevet påført større udgifter, fordi det tager længere tid at søge fuldtekst. Artiklen peger på, at udbyderne kunne støtte brugerne meget ved at bruge noget tid og nogle penge på at opbygge og vedligeholde en tesaurus.

Elliott, R. W. (1971):

Review af "A relational model of data for large shared data banks" i *Computing Reviews*, review nr.20,780 (marts 1971), p110-111. Genoptrykt i anledning af Computing Reviews's 25 års jubilæum: Computing Reviews vol.28 nr.1 (januar 1985), p61.

Elliott anerkender det solide teoretiske fundament og den elegante matematiske formulering, der ligger bag relationsdatabaserne. Han mener imidlertid, de er mere af teoretisk end af praktisk interesse. Det skyldes først og fremmest, at mange af konsistenscheckene vil være meget vanskelige at implementere og af lille praktisk værdi.

Faloutsos, Christos (1985):

"Access Methods for Text" i *Computing Surveys* vol.17 nr.1 (marts 1985), p49-74.

Faloutsos opdeler lagringsteknikkerne i fem grupper: Fuldtekstskanning, invertering, multiattribut hashing, signaturfiler og klyngeteknikkerne. De to første grupper er relevante i forbindelse med fuldtekstsøgning, og beskrivelsen af dem er kort og præcis.

Faloutsos, Christos & Chan, Raphael (1988):

"Fast Text Access Methods for Optical and Large Magnetic Disks: Designs and Performance Comparison" i (Bancilhon & DeWitt 1988), p280-293.

Artiklens udgangspunkt er, at tekstdatabaser normalt er store, og at der forekommer indsættelser i dem, men næsten aldrig sletninger og rettelser. Optiske diske, fx CD-ROM og WORM ('Write Once Read Many times'), er derfor velegnede til lagring af tekstdatabaser. Forskellige lagringsteknikkers effektivitet i forbindelse med lagring af store tekstdatabaser på optiske diske sammenlignes gennem eksperimenter.

Finkelstein, Richard & Pascal, Fabian (1988):

"SQL Database Management Systems" i *Byte* vol.13 nr.1 (januar 1988), p111-118.

Artiklen indleder med at fastslå, at der i øjeblikket kun findes seks kommercielt tilgængelige SQL-baserede relationsdatabasesystemer til PC'ere: Informix, Ingres, Oracle, SQLBase, XDB II og XQL. Artiklen fortsætter med en kort beskrivelse af de seks systemer og derefter en overfladisk test og sammenligning. Testen giver ikke anledning til, at nogle af systemerne fremhæves fremfor andre. Forfatterne konkluderer blot, at alle seks systemer har deres styrker og svagheder.

Firnberg, David (1986):

"Integrating text and non-text: its place in office automation" i (Kimberley 1986), p71-76.

Artiklen giver en oversigt over tiltag til at integrere de forskellige systemer til kontorautomation. Der skelnes mellem den fysiske integration, der er vidt fremskreden, og den logiske integration, som i vid udstrækning mangler. En af de måder, den logiske integration søges muliggjort på, er gennem standarder. I artiklen beskrives to af ISOs standarder på området: ODA (Office Document Architecture) og MOTIS, der handler om transmissionsformater for kontordokumenter.

Foskett, Douglas J. (1985):

"Thesaurus", p270-316 i Eleanor D. Dym (red.): "Subject and Information Analysis", Marcel Dekker Inc., New York (1985). 495 sider. ISBN 0-8247-7354-3.

En detaljeret gennemgang af hvad en tesaurus er og omfatter - fra ordets oprindelse til de nødvendige revisioner af tesaurussen. Artiklen handler om tesaurusser generelt, såvel dem på bogform som de edb-

baserede. Et enkelt afsnit behandler de særlige muligheder ved edb-baserede tesaurusser; men det er den 'implementationsuafhængige' del af artiklen, der er interessant.

Fox, Christopher (1989):
"A Stop List for General Text" i *SIGIR forum* (ACM's Special Interest Group on Information Retrieval) vol.24 nr.1-2 (efterår 1989/vinter 1990), p19-35.
Fox giver en opskrift på, hvordan en stopliste kan laves. Opskriften indeholder både automatiske elementer - optælling af ordfrekvenser - og manuel kontrol, der både fører til fjernelse af nogle ord og indsættelse af andre. Artiklen indeholder en stopliste på 421 ord for generel engelsk tekst.

Frøkjær, Erik & Pedersen, Gert Schmeltz (1987):
"Klar besked om ekspertsystemer", Forlaget Kommuneinformation, København. 55 sider.
Frøkjær & Pedersen giver et overblik over, hvad ekspertsystemer er, og beskriver gennem en række eksempler, hvor langt udviklingen af dem er nået. Gennem disse eksempler beskrives ekspertsystemerne snarere som systemer, der kan støtte eksperter i deres arbejde, end som systemer, der besidder eksperternes kvalifikationer. I forbindelse med ekspertsystemer indenfor det juridiske område trækkes tre punkter frem som forklaring på dette: Det første er nødvendigheden af ræsonnementer baseret på sund fornuft. Det andet er manglen på en egentlig model for juridiske afgørelser. Det tredje punkt er kompleksiteten af jura og lovgivning; denne kompleksitet skyldes blandt andet, at lovteksterne er fyldt med vagt definerede begreber.

Gauch, Susan & Smith, John B. (1989):
"An expert system for searching in full-text" i *Information Processing & Management* vol.25 nr.3, p253-263.
Artiklens kerne er den avancerede automatiske reformulering af forespørgsler, i form af udvidelser eller indsnævringer, ud fra relevansfeedback. I reformuleringen benyttes såvel automatisk behandling af bøjninger, som tesaurusrelationer og nærhedsoperatorer.

Goodman, Danny (1987):
"The complete HyperCard handbook", Bantam books, New York, Toronto. 720 sider. ISBN 0-553-34391-2.
En god og overskuelig bog om Hypercard, et af Macintosh'ens brugergrænsefladeværktøjer.

Güntzer, U.; Jüttner, G.; Seegmüller, G. & Sarre F. (1989):
"Automatic thesaurus construction by machine learning from retrieval sessions" i *Information Processing & Management* vol.25 nr.3, p265-273.
Artiklen beskriver et forsøg på at integrere vedligeholdelsen af en tesaurus i den daglige brug af et søgesystem, der bruges af flere personer. Forslag til optagelse af nye ord og etablering af relationer mellem ord kan angives i forbindelse med forespørgslerne. Systemet bruger disse forslag ligesom tesasurussens øvrige ord og relationer blot med en markering af, at der er tale om forslag. Når systemets andre brugere møder forslagene, anmoder systemet om, at de aner- eller underkender forslagene. Grundlaget for om et forslag optages eller afvises er optælling af disse aner-/underkendelser.

Harrison, Nicolas (1981):
"LEXIS: a radical approach to computer-assisted legal research" i *Program*, vol.15 nr.3 (juli 1981), p120-131.
Artiklen handler om angelsaksisk jura, og hvordan den kan drage nytte af juridiske informationssøgesystemer. Først beskrives kilderne til juridisk information (primært love, domsafsigelser og kompetente sammenskrivninger), sammenhængen mellem dem og deres placering i den juridiske sagsbehandling. Derefter beskrives LEXIS, det ene af de to store amerikanske juridiske informationssøgesystemer. LEXIS er et fuldtekstsøgesystem, og det er ifølge Harrison en radikal tilgang.

Hertzum, Morten & Søes, Henrik (1990):

"Eksperimentel systemudvikling i teori og praksis", intern rapport nr. 90-4-1 på DIKU, København. 114 sider.
Projektet består af tre dele: En teoretisk beskrivelse af eksperimentel systemudvikling, en empirisk undersøgelse af, hvordan eksperimentel systemudvikling bruges i praksis, og en analyse af fordele og begrænsninger ved eksperimentel systemudvikling. Eksperimentel systemudvikling udspringer af erkendelsen af en række problemer ved de traditionelle systemarbejdsmetoder - fasemodellerne. Disse problemer diskuteres kort.

Hoppe, Heinz Ulrich; Ammersbach, Karin; Lutes-Schaab, Barbara & Zinssmeister, Gaby (1990):
"EXPRESS: An Experimental Interface for Factual Information Retrieval" i (Vidick 1990), p63-81.
I denne artikel lægges stor vægt på tesaurussen, da den - med en analogi til ekspertsystemer - betragtes som informationssøgesystemets vidensbase. Der lokaliseres tre aktiviteter, som tesaurussen skal støtte: Den indledende formulering af forespørgslen, reformulering af forespørgslen og skimming. For at opnå den ønskede funktionalitet benyttes en facetteret tesaurus (en tesaurus der omfatter flere sammenvævede hierarkier), og tesaurussen implementeres ved hjælp af et semantisk net.

Ingwersen, Peter (1984):
"A Cognitive View of Three Selected Online Search Facilities" i *Online Review* vol.8 nr.5, p465-492.
I artiklen behandler Ingwersen blandt andet relevansfeedback. Det foreslås, at den automatisk reformulerede forespørgsel forelægges for brugeren. Så er det muligt at ændre i forespørgslen, før den udføres; den kan naturligvis også accepteres uændret.

Ingwersen, Peter (1987):
"Towards a new research paradigm in information retrieval", p150-168 i Irene Wormell (red.): "Knowledge engineering. Expert systems and information retrieval", Graham Taylor, London (1987). 182 sider.
I artiklen beskrives IR-forskningen i historisk perspektiv. Udviklingen af informationssøgesystemer karakteriseres ud fra de tre paradigmer, der har dannet grundlag for området: Det system-drevne, det bruger-orienterede og det kognitive paradigme. Synspunkterne i denne artikel støttes af Belkin & Vickery (1985).

Ingwersen, Peter & Wormell, Irene (1990):
"Informationsformidling i teori og praksis". Munksgaard, København, 104 sider. ISBN 87-16-10643-1.
En introduktion til grundbegreberne indenfor informationssøgning med specielt henblik på anvendelserne i biblioteksverdenen. Hovedvægten ligger på de historiske, kognitive og sociale aspekter - ikke på de tekniske.

ISO (1974):
"Documentation-Guidelines for the Establishment and Development of Monolingual Thesauri", ISO norm 2788, Paris.

Jaeschke, G. & Schek, Hans-Joerg (1982):
"Remarks on the Algebra of Non First Normal Form Relations", p124-138 i "Principles of database systems. Proceedings of the ACM Symposium, Los Angeles, CA, March 29-31, 1982", ACM (SIGACT-SIGMOD), New York (1982). 305 sider.
Artiklens udgangspunkt er, at udførelsen af *joins* er den mest tidskrævende operation i relationsdatabasesystemer. Mange af disse *joins* bliver nødvendige, fordi databasen for at overholde normalformerne opdeles i mange tabeller. Nogle af disse *joins* kan undgåes ved at bryde den første normalform.

Jensen, Pablo Tomas (1990):

"Hypertekst-informationssystemer", intern rapport nr. 90-9-3 (speciale) på DIKU, København. 95 sider.
Specialets teoretiske del består af et litteraturstudie, der forsøger at afgrænse og klarlægge hypertekst-begreberne, og et bud på en udviklingsmodel for hypertekstsystemer. Med udgangspunkt i denne model behandles en case: Mulighederne for og problemerne ved at konvertere Virksomheds-Karnov til et hypertekstsystem. Der udvikles en prototype på et sådant system.

Kent, William (1983):
"A simple guide to five normal forms in relational database theory" i *Communications of the ACM*, vol.26 nr.2 (februar 1983), p120-125.
Artiklen definerer og forklarer de fem normalformer, der anvendes i teorien om relationsdatabaser. Artiklens styrke er, at alle definitionerne er baseret på enkle begreber og holdt i enkle formuleringer.

Kerninghan, Brian W. & Ritchie, Dennis M. (1988):
"The C programming language", 2. udgave, Prentice Hall, Englewood Cliffs, New Jersey. 272 sider. ISBN 0-13-110362-8.
Anden udgave af den klassiske C-grundbog. God og overskuelig.

Kimberley, R. (red.) (1986):
"Integrating text with non-text: A picture is worth 1k words. Proceedings of the Institute of Information Scientists Text Retrieval '85 Conference", Taylor Graham, London. 120 sider. ISBN 0-947568-07-7.

Kleinbart, Paul (1985):
"Prolegomenon to 'intelligent' thesaurus software" i *Journal of Information Science* vol.11, p45-53.
I artiklen diskuteres, hvad tesaurussens rolle og funktion er i forbindelse med 'intelligent' informationssøgning. Derefter argumenterer Kleinbart udførligt for, at tesaurussen bedst kan udfylde denne rolle, hvis den implementeres ved hjælp af objektorienteret programmering.

Kuhn, T. S. (1962):
"The Structure of Scientific Revolutions", Chicagos Universitet, Chicago.
Her lanceres paradigmebegrebet. Kuhn brugte paradigmebegrebet til at beskrive videnskabers udvikling. Videnskabernes udvikling beskrives som en vekslen mellem normalvidenskab og krise/revolution. Normalvidenskab karaktieriseres ved, at der er ét enerådende paradigme; krise/revolution karakteriseres ved, at der er flere konkurrerende paradigmer. Senere er paradigmebegrebet blevet brugt i et utal af varierende betydninger.

Lancaster, F. Wilfrid; Rapport, R. L. & Penry, J. K. (1972):
"Evaluating the Effectiveness of an On-Line, Natural Language Retrieval System" i *Information Storage and Retrieval* vol.8, p223-245.
Lancaster m.fl. beskriver en undersøgelse af *Epilepsy Abstracts Retrieval System*. De har sammenlignet søgning på nøgleord og søgning på såvel nøgleord som resuméer (et skridt i retning af fuldtekstsøgning). De finder markant højere *recall*, når resuméerne inddrages i søgningen, og mener blandt andet, det skyldes, at brugernes sprog er i bedre overensstemmelse med det naturlige sprog i *abstract*'ene end med det sprog, nøgleordene udgør.

Leavitt, Harold J. (1964):
"Applied organization change in industry: Structural, technical and human approaches" i William W. Cooper & Harold J. Leavitt: "New perspectives in organization research", New York, p55-71.
Artiklen er en klassiker indenfor organisationsteori. Leavitt lokaliserer fire variable, der er centrale i beskrivelsen af enhver organisation, og påpeger, at de allesammen gensidigt på virker hinanden. De fire variable er: Organisationens struktur, de ansatte, opgaverne og hjælpemidlerne i form af teknologi og

andre redskaber.

Leith, Philip (1986):

"Fundamental Errors in Legal Logic Programming" i *The Computer Journal* vol.29 nr.6, p545-552.

Artiklen er et opgør med forsøgene på at reducere lovtekster til logikprogrammer i fx PROLOG. Leith påviser, at lovtekster ikke kan forstås og bruges uden at deres kontekst inddrages. Som eksempel påpeger Leith, at lovtekster også er del af en magtkamp mellem den lovgivende og den dømmende instans: Lovgiverne ønsker at styre domspraksis, og dommerne forsøger at bøje loven, så den er i overensstemmelse med deres retsopfattelse. Denne magtkamp kan direkte påvises i lovteksterne i form af såkaldte *ouster clauses*, der sigter på at give lovgiverne fuld kontrol over fortolkningen af lovteksterne.

Leith, Philip (1990):

"Formalism in AI and computer science", Ellis Horwood Limited, New York. 225 sider. ISBN 0-13-325549-2.

Leith argumenterer imod det ensidigt naturvidenskabelige perspektiv, der præger store dele af datalogien. Problemet ved det naturvidenskabelige perspektiv er, at al forståelse baseres på formalismer. Gennem en række eksempler argumenterer Leith for, at formalismer aldrig giver et dækkende billede af virkeligheden. Det er tvingende nødvendigt at inddrage sociale, psykologiske og politiske aspekter. Leith angriber specielt forsøgene på at formalisere (dele af) lovgivningen; disse forsøg er udsigtsløse af den fundamentale grund, at: *"[...] law is not about consensus, it is about control, money and power."*

Lesk, Micheal (1989):

"What to do when there's too much information" p305-318 i R. Akscyn (konferenceleder): "Hypertext '89 proceedings" (Pittsburgh, PA, 5-9 nov. 1989), ACM Press, New York, 1989. 403 sider. ISBN 0-89791-339-6.

Lesk diskuterer forskellige muligheder for at afhjælpe problemet med "too many hits". En mulighed er at præsentere passager med de hyppigst forekommende ord. Hvis databasen er opdelt i klynger, er en anden mulighed at vise såvel dokumentets titel som navnet på den klynge, dokumentet kommer fra. Generelt anbefaler Lesk interaktive løsninger fremfor detaljeret analyse af forespørgslen i et forsøg på at finde alle de relevante dokumenter - og kun dem - første gang.

Lindgreen, Paul (1988):

"Analyse af organisatoriske systemer" (forelæsningsnoter til Informatik 1), Institut for Informatik og Økonomistyring, Handelshøjskolen, København. 166 sider.

Bogen handler om grundlaget for og udviklingen af et detaljeret værktøj til analyse af aktiviteterne i organisationer. Bogen er efter vores mening dårlig, men indeholder enkelte gode afsnit med grundige diskussioner af nogle centrale begreber, fx data, information og viden.

Luhn, Hans Peter (1957):
"A statistical approach to mechanized encoding and searching of literary information" i *IBM Journal of Research and Development* vol.1 nr.4 (oktober 1957), p309-317.
I denne artikel foreslår Luhn som den første, at afgørelsen af hvilke dokumenter, der er relevante for en given forespørgsel, skal træffes ved sammenligning af statistisk bestemte indholdsidentifikatorer. Identifikatorerne - nøgleordene - skal bestemmes ved optælling af frekvenserne for de ord, der forekommer i dokumenterne.

Luhn, Hans Peter (1958):
"The automatic creation of literature abstracts" i *IBM Journal of Research and Development* vol.2, p159-165.
I denne artikel introduceres ideen om en øvre og en nedre grænse for gode nøgleords frekvens. Ordene med lav frekvens er dårlige nøgleord; de er for specielle og vil sjældent blive brugt i forespørgsler. Ordene med høj frekvens er ligeledes dårlige nøgleord; de er for almindelige og siger ikke noget om dokumentets indhold. Ordene med frekvenser i mellemområdet er derimod meningsbærende og gode nøgleord.

Lynch, Clifford A. & Stonebraker, Michael (1988):
"Extended User-defined Indexing with Application to Textual Databases" i (Bancilhon & DeWitt 1988), p306-317.
Den mest tidskrævende operation i RDBMS er *join*-operationen. I artiklen diskuteres mulighederne for at effektivisere RDBMS ved at erstatte nogle *joins* med abstrakte datatyper og brugerdefinerede operatorer. Diskussionen er specielt rettet mod lagring af store tekstmængder ved hjælp af RDBMS. Lynch & Stonebraker eksperimenterer med flere forskellige typer indeks, der skal støtte operatorerne og de abstrakte datatyper. Deres eksperimenter indikerer, at både plads- og tidsforbruget er betydeligt mindre med operatorer og abstrakte datatyper end med *joins*.

Maarek, Yoëlle S. & Smadja, Frank A. (1989):
"Full Text Indexing Based on Lexical Relations. An Application: Software Libraries" i (Belkin & van Rijsbergen 1989), p198-206.
Artiklen handler om de specielle forhold, der gør sig gældende, når søgesystemer anvendes i forbindelse med programbiblioteker. Det udviklede søgesystem er baseret på automatisk indeksering ud fra programmernes fulde tekst.

Mackay, D. M. (1960):
"What makes the question?" i The Listener vol.63, p789-790.
Artiklen markerer det første forsøg på at undersøge, hvad der får mennesker til at stille forespørgsler til søgesystemer.

Marchionini, Gary (1989):
"Making the transition from print to electronic encyclopaedias: adaption of mental models" i *International Journal of Man-Machine Studies* vol.30 nr.6 (juni 1989), p591-618.
Artiklen handler om forskellen på at bruge elektroniske ordbøger/leksika og papirudgaverne af de samme værker. Specielt undersøges det empirisk, hvor meget kendskabet til ordbøger/leksika på papirform påvirker brugen af elektroniske søgesystemer. Derudover gives en detaljeret beskrivelse af, hvilke søgefaciliteter brugerne anvendte og hvordan. Formålet med undersøgelsen er at prøve at afdække nogle af de metaforer - mentale modeller -, brugerne baserer deres forståelse af søgesystemerne på.

Matos, Victor M. & Jalics, Paul J. (1989):
"An Experimental Analysis Of The Performance Of Fourth Generation Tools On PCs" i *Communications of the ACM*, vol.32 nr.11 (November 1989), p1340-1351.
I artiklen sammenlignes svartiderne for forskellige relationsdatabasesystemer til PC'ere gennem en række eksperimenter. Matos & Jalics finder, at Oracle er meget langsommere end de fleste andre

relationsdatabasesystemer, specielt når det gælder *joins* af store tabeller. I et af deres eksperimenter er Oracle godt 6 timer om en *join*, som i Paradox udføres på 15 sekunder.

McMath, Charles F.; Tamaru, Robert S. & Rada, Roy (1989):
"A graphical thesaurus-based information retrieval system" i *International Journal of Man-Machine Studies* vol.31 nr.2 (august 1989), p121-147.
I artiklens første del gøres rede for mulighederne for at ræsonnere på relationerne mellem begreber ved hjælp af en tesaurus. Kernen i dette er det afstandsmål, der bruges til at måle 'den semantiske afstand' mellem to begreber. I artiklens anden del arbejdes med udviklingen af en grafisk brugergrænseflade, specielt en grafisk præsentation af tesaurussen. Systemet styres ved hjælp af mus og 'klik'.

Mili, Hafedh & Rada, Roy (1988):
"Merging Thesauri: Principles and Evaluation" i *IEEE Transactions on Pattern Analysis and Machine Intelligence* vol.10 nr.2 (marts 1988), p204-220.
I artiklen behandles mulighederne for automatisk at slå to delvist overlappende tesaurusser sammen. Der er to centrale problemer i dette: For det første at finde de par af begreber (et fra hver tesaurus), som er tilstrækkeligt ens til, at de skal slås sammen. For det andet at 'den semantiske afstand' mellem naboniveauer formodentlig er forskellig såvel fra niveau til niveau som mellem de to tesaurusser. Kernen i løsningen af begge problemer er udviklingen af et mål for 'den semantiske afstand' mellem to begreber.

Mintzberg, Henry (1983):
"Structures in fives: Designing effective organizations", Prentice-Hall, Englewood Cliffs, New Jersey. 312 sider. ISBN 0-13-854191-4.
Mintzberg beskriver fem organisationstyper: Den enkle organisation, maskinbureaukratiet, fagbureaukratiet, den divisionaliserede organisation og adhokratiet. Han påstår, at hvis man inddrager de mulige blandingsformer, dækker disse fem organisationstyper i øjeblikket de i praksis forekommende organisationer. Advokatkontoret og universitetet bruges som typeeksempler på fagbureaukratiet. Fagbureaukratiet kendetegnes ved, at den dominerende koordinationsmekanisme er standardisering af de ansattes færdigheder (i modsætning til fx overvågning eller standardisering af arbejdsprocessen), og den centrale medarbejdergruppe er de menige ansatte (i modsætning til fx ledelsen eller mellemlederne).

Naur, Peter (1965):
"The place of programming in a world of problems, tools and people", side 195-199 i Wayne A. Kalenich (red.): "Proceedings of the IFIP Congress 65", vol.1 (New York, maj 1965), Spartan Books, Washington DC. 304 sider.
Artiklen handler om den situation, ethvert edb-system indgår i: Brugssituationen. Brugssituationen beskrives som et samspil mellem tre elementer: Mennesker, opgaver og hjælpemidler. Naur understreger, at der er tale om et samspil: De forhåndenværende hjælpemidler påvirker personernes opfattelse af opgaverne, og personernes forståelse af opgaverne er bestemmende for deres mening om, hvad der er et velegnet eller ønskeligt hjælpemiddel. Efter en generel diskussion specialiseres situationen til de tilfælde, hvor hjælpemidlet er et edb-system.

O'Connor, John (1975):
"Retrieval of answer-sentences and answer-figures from papers by text searching" i *Information Processing & Management* vol.11, p155-164.
I artiklen behandles passage-præsentation, altså præsentation af passager fra teksten fremfor blot dokumenternes titler og lignende. Passage-præsentation gør både brugernes adgang til den relevante information og mulighederne for afvisning af de irrelevante, men fremfundne, dokumenter lettere og hurtigere.

Oddy, R. N.; Robertson, S. E.; van Rijsbergen, C. J. & Williams, P. W. (red.) (1981):
"Information Retrieval Research". Proceedings from the symposium Research and Development in Information Retrieval at St. John's College, Cambridge, in June 1980. The first joint BCS and ACM symposium in Information storage and retrieval. Butterworth, London, 389 sider. ISBN 0-408-10775-8.
Konference-*proceedings*. Denne konference er forløberen for den årlige konference, der afholdes under

overskriften *Research and Development in Information Retrieval*.

Palay, Andrew J. & Fox, Mark S. (1981):
"Browsing through databases" i (Oddy m.fl. 1981), p310-324.
I artiklen beskrives skimming ud fra BROWSE/ZOG-systemet. Det pointeres, at skimming ikke er et alternativ, men et supplement til andre søgeteknikker. Ex: Hvis en bruger ved præcis, hvad han søger, skal han ikke være nødt til at bevæge sig gennem netværket af dokumenter og referencer for at komme til det.

Rada, Roy & Martin, Brian K. (1987):
"Augmenting Thesauri for Information Systems" i *ACM Transactions on Office Information Systems* vol.5 nr.4 (oktober 1987), p378-392.
Artiklen består af en indledende diskussion af mulighederne og behovet for at udvikle systemer til at støtte eksperter i beslutningerne om med hvilke ord og relationer mellem ordene, en tesaurus skal udvides. Den anvendte tesaurus implementeres ved hjælp af en relationsdatabase. Det arbejde, der startes med denne artiklen, fortsætter i (Mili & Rada 1988) og (McMath m.fl. 1989).

Remmen, Arne (1989):
"Lapperne på knapperne", interview med Arne Remmen foretaget af Jes Michelsen, *DJØF-bladet*, nr.24 (8. december 1989), p20-22.
Remmen giver udtryk for, at jurister har behov for væsentligt mere uddannelse og viden om edb. En af årsagerne til det er efter Remmens mening, at jurister fokuserer for meget på avancerede edb-systemer, som endnu kun er under udvikling. De glemmer at interessere sig for de almindelige edb-systemer og overser dermed, at det er på dette område, den store udvikling vil ske: Tekstbehandling, elektronisk post, edb-arkivering osv. Remmen mener, de systemer, der vil vinde frem, er dem, der peger hen imod den integrerede arbejdsplads.

Retsinformation (1989):
"RETSINFORMATION Sådan gør man" Justitsministeriet, Sekretariatet for Retsinformation, Schultz Forlag, København. 29 sider.
Hæftet er en minivejledning i brugen af Retsinformation. Det beskriver Retsinformation generelt, hvilke lovtekster systemet omfatter, og hvordan de er organiseret. Derefter redegøres for, hvordan man slutter sig til systemet, og de seks vigtigste søgekommandoer gennemgås kort.

Salton, Gerard (1986):
"Another look at automatic text-retrieval systems" i *Communications of the ACM* vol.29 nr.7 (juli 1986), p648-656.
Denne artikel indeholder dels en kritik af konklusionerne i (Blair & Maron 1985) dels en 9-punktsplan for, hvordan et '*state of the art*'-søgesystem til nøgleordsbaseret søgning kan opbygges. Kritikken af Blair & Maron er først og fremmest, at der ikke er noget usædvanligt i at få et *recall* på ca 20% sammen med en *precision* på ca 80%. Salton mener, det er velkendt, at der skal bredere forespørgsler - lavere *precision* - til at få det ønskede højere *recall*.

Salton, Gerard & Buckley, Christopher (1988):
"Term-weighting approaches in automatic text retrieval" i *Information Processing & Management* vol.24 nr.5, p513-523.
Artiklen giver en kort og præcis beskrivelse af term-vægtning - en partiel match søgeteknik, hvor dokumenterne og forespørgslerne beskrives ved vektorer i det vektorrum, der udspændes af alle de potentielle nøgleord. Term-vægtning er baseret på et udtryk for forespørgsel/dokumentligheden. Der benyttes flere forskellige udtryk for denne lighed; de er alle baseret på tre størrelser: Term-frekvens, omvendt term-frekvens og term-relevans.

Salton, Gerard; Fox, Edward A. & Wu, Harry (1983):
"Extended Boolean Information Retrieval" i *Communications of the ACM* vol.26 nr.12 (december 1983), p1022-1036.

I artiklen defineres, beskrives og eksperimenteres med udvidet boolsk søgning, der er en kombination af term-vægtning og boolsk søgning. Udvidelsen i forhold til boolsk søgning består i, at såvel ordene i dokumenterne som ordene i forespørgslerne og de boolske operatorer kan tildeles vægte.

Salton, Gerard & McGill, Micheal J. (1983):
"Introduction to Modern Information Retrieval". McGraw-Hill, New York, 448 sider. ISBN 0-07-054484-0.
En grundbog i informationssøgning med hovedvægten på nøgleordsbaseret søgning.

Salton, Gerard & Schneider, Hans-Jochen (red.) (1983):
"Research & Development in Information Retrieval" (proceedings, Berlin, Maj 1982), Lecture Notes in Computer Science nr.146, Springer-Verlag, Berlin Heidelberg, 311 sider.
Proceedings fra 1982-udgaven af den årlige hovedkonference om forskningen i informationssøgning.

Salton, Gerard; Yang, C. S. & Yu, C. T. (1975):
"A Theory of Term Importance in Automatic Text Analysis" i *Journal of the American Society for Information Science* (ASIS) vol.26 nr.1, p33-44.
Artiklen handler om teknikker til automatisk bestemmelse af nøgleord. Ord, der kun forekommer enkelte gange i et dokument, og ord, der forekommer virkelig mange gange, er dårlge nøgleord. Der antages således at være en øvre og en nedre grænse for frekvensen af gode nøgleord. I artiklen foreslås en frekvens på 1% af dokumenterne som nedre grænse og 10% som øvre grænse.

Smeaton, A. F. & van Rijsbergen, C. J. (1986):
"Information retrieval in an office filing facility and future work in project Minstrel" i *Information Processing & Management*, vol.22 nr.2, p135-149.
Artiklen giver - med fokus på kontorverdenen - en oversigt over søge- og arkiveringssystemer til ustruktureret tekst, lyd og billeder. Det er dog primært mulighederne i forbindelse med tekst, artiklen behandler. Specielt udskilles tre væsensforskellige typer søgninger: Associationsbaseret, indholdsbaseret og klassifikationsbaseret. Et søge- og arkiveringssystem bør støtte alle disse tre typer søgninger - såvel når det gælder tekst, som når det gælder lyd og billeder.

Smith, Linda C. (1981):
"'Memex' as an image of potentiality in information retrieval research and development" i (Oddy m.fl. 1981), p345-369.
I Artiklen diskuteres den betydning, Bush's artikel "As we may think" (Bush 1945) har haft for IR-forskningen. Smith karakteriserer Memex, det system Bush beskriver, som 'an image of potentiality' og lokaliserer dens indflydelse til fem brede områder: Historisk perspektiv, hardware, lagring af information, software og 'Personal Information Systems'.

Smith, Linda C. (1983):
"Machine intelligence vs. machine-aided intelligence in information retrieval: A historical perspective" i (Salton & Schneider 1983), 263-274.
I artiklen diskuteres forskellen på to niveauer af automatisering: Maskinintelligens og maskinstøttet intelligens. Indenfor IR var denne problemstilling ny i 1983. Smith trækker på de erfaringer, der er opnået i forbindelse med automatisk oversættelse. Indenfor dette område har problemstillingen været aktuel længe, og automatisk oversættelse har flere væsentlige lighedspunkter med IR.

Soergel, Dagobert (1974):
"Indexing Languages and Thesauri: Construction and Maintenance", John Wiley & Sons, Los Angeles, California, 632 sider. ISBN 0-471-81047-9.
En meget omfattende og grundig gennemgang af, hvordan tesaurusser konstrueres og vedligeholdes. Bogen behandler tesaurusser generelt, dvs både dem på bogform og de edb-baserede. Bogen er let at finde rundt i, så man behøver ikke læse alle 632 sider.

Sparck Jones, Karen (1973):
"Does indexing exhaustivity matter?" i *Journal of the American Society for Information Science* (ASIS) vol.24 nr.3, p313-316.
Indekseringsglosarier beskrives normalt ved hjælp af to mål: Bredde og dybde. Disse mål er meget svære at kvantificere. Sparck Jones foreslår, at bredden er relateret til antallet af nøgleord, der hæftes på et givet dokument, og at dybden er relateret til antallet af dokumenter, der får et givet nøgleord hæftet på sig - få dokumenter svarer til stor dybde.

Sparck Jones, Karen (1974):
"Automatic indexing" i *Journal of Documentation* vol.30 nr.4 (december 1974), p393-432.
Oversigt over resultaterne indenfor det system-drevne paradigme i perioden 1968-73. Det er bemærkelsesværdigt så mange af dette paradigmes resultater, der allerede er nået på dette tidspunkt.

Sprowl, James A. (1981):
"WESTLAW vs. LEXIS: Computer-Assisted Legal Research Comes of Age" i *Program* vol.15 nr.3 (juli 1981), p132-141.
Artiklen beskriver de to store amerikanske juridiske fuldtekstsøgesystemer, LEXIS og WESTLAW. WESTLAW omfatter både fuldtekst og visse faciliteter til nøgleordsbaseret søgning. Sprowl mener, at kombinationen af fuldtekstsøgning og blot en grov indeksering med nøgleord er en fordel, da nøgleordene så kan bruges til at dele databasen op i klynger. Det giver mulighed for at afgrænse søgningerne til en del af databasen.

Strong, Gary W. & Drott, M. Carl (1986):
"A thesaurus for end-user indexing and retrieval" i *Information Processing & Management* vol.22 nr.6, p487-492.
I artiklen argumenteres for, at hvis et informationssøgesystem skal være effektivt, så skal det omfatte en tesaurus, der giver brugerne mulighed for at orientere sig i fagområdets begrebsapparat før søgningen udføres. Strong & Drott mener, en facetteret tesaurus (en tesaurus der omfatter flere sammenvævede hierarkier) bedst opfylder dette krav. Det fastslås, at vedligeholdelsen af tesaurussen kræver central styring.

Tenopir, Carol (1984):
"Full-Text Databases", p215-246 i Martha E. Williams (red.): "Annual Review of Information Science and Technology" (ARIST) vol.19, American Society for Information Science (ASIS).
En oversigtsartikel over fuldtekstdatabaser og fuldtekstsøgning. Tenopir beskriver eksisterende anvendelser af fuldtekstsøgesystemer indenfor flere forskellige områder: Jura, nyhedsformidling og biblioteksverdenen. Artiklens hovedafsnit handler om systemernes søgeeffektivitet. Det fremgår, at forskerne er nogenlunde enige om, at fuldtekstsøgning giver højere *recall* end nøgleordsbaseret søgning, mens *precision* er ret ens. Enkelte finder, at nøgleordsbaseret søgning giver højere *precision* end fuldtekstsøgning. Det konstateres endvidere, at ingen søgeteknik er tilstrækkelig i sig selv - med hver eneste søgeteknik findes dokumenter, som ikke findes med de andre.

Tenopir, Carol & Ro, Jung Soon (1990):
"Full text databases", Greenwood Press, New York, 251 sider. ISBN 0-313-26303-5.
Bogen søger at give et overblik over fuldtekstsøgesystemer. Den indledes med en oversigt over, hvor fuldtekstsøgesystemer anvendes, og en detaljeret gennemgang af, hvilke faciliteter der typisk findes i et fuldtekstsøgesystem. I bogens hovedkapitler behandles en række eksperimenter, hvor fuldtekstsøgning sammenlignes med nøgleordsbaseret søgning, og forskellige faciliteter til fuldtekstsøgning sammenlignes indbyrdes. De forskellige faciliteter til fuldtekstsøgning er primært forskellige nærhedsoperatorer.

Thompson, R. H. & Croft, W. Bruce (1989):

"Support for browsing in an intelligent text retrieval system" i *International Journal of Man-Machine Studies* vol.30 nr.6 (juni 1989), p639-668.

Thompson & Croft diskuterer skimming ud fra skimmefacaliteterne i I^3R-systemet (I^3R står for 'Intelligent Interface for Information Retrieval'). Skimming kan ikke erstatte de øvrige søgeteknikker, men supplere dem. Styrken ved skimming er, at brugeren ikke behøver formulere den utilstrækkelige viden før søgningen starter. Brugeren skal i stedet genkende/opdage relevant information under skimmingen - noget mennesker har meget lettere ved. Derudover giver skimming mulighed for at følge forskellige fastlagte typer associationer, fx: Hvis dette dokument er interessant, er nogle af dets henvisninger så også interessante?

Thorup, Sju Georgine; Broløs, Jette Holm; Kjeldsen, Jon; Dadsetan, Farzin & Pilgaard, Kristian Bang (1990):

"Undersøgelse af hypertekstværktøjer", intern rapport nr. 90-10-5 på DIKU, København. 99 sider.

Projektet indeholder en beskrivelse af de forskellige typer knuder og referencer, der kan indgå i et søgesystem baseret på skimming. Derudover diskuteres baggrunden for samt fordele og ulemper ved skimming.

van Rijsbergen, C. J. (1979):

"Information retrieval", 2. udgave, Butterworth, London, 208 sider. ISBN 0-408-70929-4.

En grundbog i informationssøgning med hovedvægten på klynge-teknikkerne og de probabilistiske teknikker. Bogen er en god introduktion også i forbindelse med fuldtekstsøgning.

Vidick, Jean-Luc (red.) (1990):

"13th International Conference on Research and Development in Information Retrieval" (Bruxelles, Belgien, 5-7 september 1990), ACM, New York, 509 sider. ISBN 0-89791-408-2.

Proceedings fra 1990-udgaven af den årlige hovedkonference om forskningen i informationssøgning.

von Eyben, W. E. (1989):

"Karnovs Lovsamling", en ajourført version af en artikel i "Danske Opslagsværker" (1977), p10-18.

Artiklen gør rede for den udvikling Karnovs Lovsamling har gennemgået siden *"Hvermands Lovbog"*, forløberen for Karnovs Lovsamling, udkom i 1924. De tre elementer i lovsamlingen - lovteksterne, noterne og registrene - beskrives grundigt, og arbejdsgangen i ajourføringen gennemgås.

Willett, Peter (1988):

"Recent trends in hierarchic document clustering: A critical review" i *Information Processing & Management* vol.24 nr.5, p577-597.

En meget detaljeret oversigtsartikel over klynge-teknikkerne.

Winograd, Terry & Flores Fernando (1986):

"Understanding Computers and Cognition: A New Foundation for Design", Ablex Publishing Corporation, Norwood, New Jersey. 207 sider. ISBN 0-89391-050-3.

I denne bog gør Winograd & Flores op med de antagelser, AI-forskningen bygger på. Grundlaget for dette opgør er først og fremmest Heideggers fænomenologi og en sprogteori, der bestemmer sproget som handling - talehandlingsteorien. Bogen består af tre dele: Først opstiller Winograd & Flores et teoretisk grundlag for forståelse af menneskers handlinger. Dernæst bruges dette grundlag som udgangspunkt for en fundamental og overbevisende afvisning af AI-forskningens muligheder for at realisere sine mål. Endelig prøver Winograd & Flores at skitsere fundamentet for en ny tilgang til design af edb-systemer baseret på deres teoretiske grundlag. Bogen er også interessant i lyset af, at Winograd er en af de forskere, der har lavet noget det mest anerkendte arbejde indenfor AI, nemlig SHRDLU-systemet.

Bilag 1: Ordliste

Ordlisten omfatter 55 af specialets centrale begreber. Vi har gennem hele specialet brugt danske termer. Det er derfor de danske termer, der er grundlaget for ordningen af ordlisten. Den engelske term er - hvis en sådan findes - angivet i parentes efter den danske.

Afskæring ('stemming', 'truncation'). Afskæring er en teknik, der bruges til automatisk behandling af bøjninger (se også dette). Ved afskæring reduceres ordene til deres grundform ved at fjerne endelser, der indgår i en prædefineret tabel over endelser. Afskæringsteknikkerne spænder fra det meget simple til komplicerede algoritmer, der foretager afskæringen i flere trin, ex: Først genitivendelser, så flertalsendelser osv.

Begrebsbaseret fremfinding ('conceptor-based retrieval'). Begrebsbaseret fremfinding er en overbygning på boolsk søgning, hvor brugeren ikke behøver bekymre sig om parenteser og boolske operatorer. Kernen i begrebsbaseret fremfinding er, at forespørgslen formuleres som en fællesmængde af begreber. Hvert begreb beskrives ved en række ord; efter hver beskrivelse af et begreb angiver brugeren, om forespørgslen er slut eller omfatter flere begreber.

Boolsk søgning ('boolean retrieval'). Boolsk søgning er en søgeteknik, hvor forespørgslerne består af søgeord knyttet sammen med de boolske operatorer (OG, ELLER og IKKE). Boolsk søgning er en eksakt match teknik. Den karakteriseres ved ikke at omfatte vægtning (hverken af søgeordene, de boolske operatorer eller ordene i dokumenterne), ved ikke at rangordne de fremfundne dokumenter og ved kun at muliggøre søgn ing på enkeltord. Se også nærhedsoperatorer.

Bredde af indekseringsglosarium ('exhaustivity'). Mål for et indekseringsglosariums effektivitet: Det antal forskellige emner, der kan indekseres med glosariet. Se også dybde af indekseringsglosarium.

Bruger-orienterede paradigme ('the user modelling paradigm'). Det bruger-orienterede paradigme indenfor forskningen i informationssøgning opstår i løbet af 70'erne, som en reaktion på det system-drevne paradigmes tekniske fokus. Med det bruger-orienterede paradigme sættes fokus på informationssøgningens psykologiske og sociologiske aspekter, specielt på brugerens problem: Formuleringen af forespørgslen ud fra en tilstand med utilstrækkelig viden.

Bøjninger, automatisk behandling af. Automatisk behandling af bøjninger går ud på at reducere ord til deres grundform. Bøjninger kan behandles ved hjælp af regler eller opslag. Regler er mest udbredt og omfatter en lang række algoritmer til afskæring. Med regler fokuseres på de svagt bøjede ord med regelrette endelser, mens de stærkt bøjede ord ikke reduceres til deres grundform eller reduceres til en forkert grundform. Med opslag baseres behandlingen af bøjninger på omfattende ordbøger over grundformer og de tilhørende bøjninger. Problemet med opslag er, at dokumenterne meget ofte indeholder ord, der ikke står i ordbogen. Se også afskæring.

Data. Data er konkrete fysiske fænomener, fx dokumenterne i et fuldtektsøgesystem. Data kan lagres, behandles og transmitteres af såvel mennesker som datamaskiner. Se også information og viden.

Dybde af indekseringsglosarium ('specificity'). Mål for et indekseringsglosariums effektivitet: Glosariets evne til at beskrive de emner, dokumenterne omhandler, præcist. Se også bredde af indekseringsglosarium.

Edb-Karnov. Vores prototype på et fuldtekstsøgesystem til fagfolk. Navnet skyldes, at datagrundlaget for prototypen er godt 400 sider fra Virksomheds-Karnov.

Eksakt match. Eksakt match betegner en gruppe søgeteknikker med boolsk søgning som den mest udbredte. Eksakt match teknikkerne kendetegnes ved blot at opdele dokumenterne i to grupper: De fremfundne og de ikke-fremfundne. Der foretages således ingen rangordning af de fremfundne dokumenter, ligesom der ikke er nogen mulighed for at vægte søgeordene eller ordene i dokumenterne.

Fagfolk ('professionals'). Fagfolk adskiller sig fra såvel personer uden faglig kompetence som personer med mere rutineprægede opgaver. Fagfolk har stor frihed i udøvelsen af deres arbejde. De er ikke underlagt en stram styring og kontrol; deres indsats bestemmes primært af deres personlige engagement og ansvarlighed. Fagfolks arbejdsopgaver er komplicerede og kan ikke reduceres til anvendelse af regler. En væsentlig del af fagfolks kompetence er intuition og sund fornuft opnået gennem erfaring. Jurister er et godt eksempel på fagfolk.

Forespørgsel ('query'). Forespørgsler udspringer en erkendelse af en tilstand af utilstrækkelig viden. Den utilstrækkelige viden formaliseres og konkretiseres til et informationsbehov, der videre formuleres og repræsenteres i en forespørgsel. Forespørgsler formuleres i søgesystemets forespørgselssprog, som er bestemt af søgeteknikken.

Fritekst-. Se fuldtekst-.

Fuldtekstrepræsentation. Fuldtekstrepræsentation er den tekstrepræsentation, der benyttes i fuldtekstsøgesystemer. Ved fuldtekstrepræsentation findes hele teksten i søgesystemet. Fuldtekstrepræsentation står i modsætning til indeksering.

Fuldtekstsøgesystem. I dette projekt bruges følgende definition: Fuldtekstsøgesystemer er den type informationssøgesystemer, hvor dokumenternes fulde tekst (eksklusiv eventuelle figurer, billeder og lignende) er til rådighed i systemet, og søgning foregår i hele teksten.

Fuldtekstsøgning ('full text retrieval') Fuldtekstsøgning betegner i dette projekt søgning, hvor dokumenternes fulde tekst (eksklusiv eventuelle figurer, billeder og lignende) er til rådighed, og søgning foregår i hele teksten.

Fuzzy mængder ('fuzzy sets'). Fuzzy mængder er en partielt match søgeteknik og et af forsøgene på at udvide boolsk søgning. Fuzzy mængder stammer fra matematik og betegner en type mængder, hvor medlemsfunktionen kan antage alle værdier i intervallet fra ikke-medlem til medlem, ikke blot de to yderværdier. Med dette mængdebegreb fås umiddelbart en partielt match søgeteknik, der ligger i forlængelse af boolsk søgning.

Homografer. To ord er homografer, hvis de staves ens, men iøvrigt er forskellige. Homografer er et problem i forbindelse med fuldtekstsøgning, da de giver anledning til fremfinding af irrelevante dokumenter.

Indeksering. Indeksering består i at repræsentere et dokument ved en præcis og specifik gruppe af nøgleord. Nøgleordene behøver ikke forekomme i dokumentet; de vælges typisk fra en tesaurus eller et andet kontrolleret glosarium. Indeksering kan foregå manuelt/intellektuelt ved, at en indeksør gennemlæser dokumentet og afgør, hvilke nøgleord der er relevante, eller automatisk ud fra optællinger af ordfrekvenser. Søgning i en indekseret dokumentsamling kaldes nøgleordsbaseret søgning.

Information. Information er et abstrakt fænomen og eksisterer kun under kommunikation som en overgangsform mellem viden og data. En del af en persons samlede viden kan formaliseres og derved blive til kommunikerbar information. Information kan repræsenteres og derved blive til data, der kan lagres, transmitteres osv. Data kan afkodes

af dem, der kender koden, derved får data mening og bliver til information. Endelig kan information erkendes og derved blive en del af en persons viden. Se også data og viden.

Informationssøgesystem ('Information retrieval system', 'IR-system' eller blot 'IRS'). Fællesbetegnelse for de søgesystemer, der giver brugeren mulighed for at søge i en dokument- eller tekstsamling. Informationssøgesystemer omfatter såvel fuldtekstsøgesystemer som systemer, der tilbyder nøgleordsbaseret søgning.

Informationssøgning ('information retrieval' eller blot 'IR'). Fællesbetegnelse for de forskellige typer søgning i dokument- og tekstsamlinger. Informationssøgning omfatter såvel fuldtekstsøgning som nøgleordsbaseret søgning.
Inverteret fil ('inverted file'). En inverteret fil dannes ud fra en tekstfil - et dokument. Den inverterede fil består af en indgang for hvert ord i teksten og hægter til alle ordets forekomster i teksten. En inverteret fil implementeres typisk ved hjælp af to filer: En indeksfil og en forekomstfil. Indeksfilen indeholder de inverterede ord og peger ind i forekomstfilen; forekomstfilen angiver ordenes placering i teksten. Se også stopliste.

Invertering. Invertering er den proces, der omfatter etableringen af en inverteret fil ud fra et eller flere dokumenter. Vi benytter primært ordet i betydningen invertering af hele teksten; men inverteringen kan begrænses til invertering af nøgleord. Se også inverteret fil og stop-liste.

Joker-operator ('wildcard'). En joker-operator er et tegn, der kan placeres i forespørgslens søgeord som erstatning for et vilkårligt tegn eller en vilkårlig gruppe af tegn. Joker-operatorer bruges typisk til at gøre søgningerne ufølsomme overfor variationer i søgeordenes bøjningsformer. Ex: Søgeordet 'operator%' matcher alle ord, der har 'operator' som prefix.

Juridisk sagsbehandling. Juridisk sagsbehandling betegner det til enhver tid eksisterende samspil mellem juristen, sagen og hjælpemidlerne i form af lovsamlinger og lignende. Juridisk sagsbehandling er et komplekst område og lader sig ikke reducere til anvendelse af regler. En væsentlig del af juristers kompetence er således intuition og sund fornuft opnået gennem erfaring.

Klynger, opdeling af databasen i. Opdeling af databaser i klynger er baseret på de veletablerede indekserings- og klyngeteknikker, men er endnu ret nyt i forbindelse med fuldtekstsøgesystemer. Formålet med opdeling af databasen i klynger er at give mulighed for at begrænse søgningen til en del af databasen. Opdelingen skal altså gruppere dokumenterne i en række klynger, der hver omfatter et (bredt) emne. En sådan opdeling vil ofte være et hierarki, hvor hvert dokument placeres i én klynge; men det behøver ikke være tilfældet.

Klyngeteknikker ('clustering techniques', 'clustering methods' eller blot 'clustering'). Fællesbetegnelse for en gruppe tekstrepræsentations- og søgeteknikker. Hovedideen er at samle dokumenter, der ligner hinanden, i klynger, samle disse klynger i overklynger osv. Sammenligningen af dokumenterne foregår ved hjælp af nøgleord og partielt match. En ny og meget detaljeret oversigtsartikel over klyngeteknikkerne er (Willett 1988).

Kognitive paradigme ('the cognitive paradigm'). Det kognitive paradigme indenfor forskningen i informationssøgning opstår midt i 80'erne. Det markerer, dels at forskere fra

det system-drevne og fra det bruger-orienterede paradigme begynder at interessere sig for hinandens resultater, dels at forskerne begynder at benytte AI-teknikker i konstruktionen af informationssøgesystemer. Med det kognitive paradigme fokuseres ikke længere på en udvalgt del af informationssøgningen, men på hele situationen med en understregning af, at der er tale om en kognitiv proces.

Lagringsteknikker. Lagringsteknikker er de teknikker, der bruges til at lagre dokumenternes tekst i søgesystemet. Valget af lagringsteknik afhænger i meget høj grad af hvilken funktionalitet søgesystemet skal have. I dette projekt behandles tre forskellige lagringsteknikker: Sekventielle filer, inverterede filer og relationsdatabaser.

Lovtekst. Lovtekst bruges i dette projekt som samlebetegnelse for love, lovbekendtgørelser, bekendtgørelser og cirkulærer.

Nøgleordsbaseret søgning. Ved nøgleordsbaseret søgning indekseres hvert dokument ved en gruppe nøgleord, og søgningerne omfatter udelukkende disse nøgleord. Den gruppe nøgleord, et dokument er indekseret med, udgør en fortolkning af dokumentet med vægten lagt på dokumentets hovedlinier.

Nærhedsoperatorer ('proximity measures'). Nærhedsoperatorer er en udbredt udvidelse af boolsk søgning. Nærhedsoperatorer indgår i forespørgslerne. De bruges til at foreskrive, at to ord ikke blot skal forekomme i samme dokument, men i samme afsnit, samme sætning eller lignende. Den nærhedsoperator, der foreskriver, at de to ord forekommer ved siden af hinanden, giver mulighed for søgning på vendinger. Ex: 'eksakt NABO match' finder et dokument frem, hvis vendingen 'eksakt match' forekommer i dokumentet.

Paradigme. Paradigmebegrebet blev lanceret af Kuhn (1962) og betyder: Et grundmønster for videnskabsudøvelse, der har en sådan kvalitet, at det formår at tilstrække og samle en forskergeneration, som derved kommer til at udgøre en videnskabssamfund. Kuhn brugte paradigmebegrebet til at beskrive videnskabers udvikling. Han beskriver videnskabernes udvikling som en vekslen mellem normalvidenskab og krise/revolution. Normalvidenskab karaktieriseres ved, at der er ét enerådende paradigme. Krise/revolution karakteriseres ved, at der er flere konkurrerende paradigmer. Senere er paradigmebegrebet blevet brugt i et utal af varierende betydninger.

Partielt match. Partielt match betegner en stor gruppe søgeteknikker, der tillader vægtning af såvel søgeordene som ordene i dokumenterne. Da partielt match teknikkerne arbejder med vægte, bliver det muligt at rangordne de fremfundne dokumenter. Udvidet boolsk søgning og term-vægtning er eksempler på partielt match søgeteknikker.

Passage-præsentation ('passage retrieval'). Ved passage-præsentation vises ikke blot de fremfundne dokumenters titler og lignende oplysninger, men også de passager i dokumenterne, hvor søgeordene forekommer. Passager giver brugeren væsentligt bedre mulighed for at vurdere dokumenternes relevans end bibliografiske oplysninger alene, idet søgeordenes kontekst inddrages i præsentationen.

Precision. Mål for søgeeffektiviteten: Den brøkdel af de fremfundne dokumenter, der er relevante. Høj *precision* betyder, at størstedelen af de fremfundne dokumenter er relevante. *Precision* siger derimod ikke noget om, hvor mange relevante dokumenter der ikke er blevet fundet frem. Se også *recall*.

Recall. Mål for søgeeffektiviteten: Den brøkdel af de relevante dokumenter, der findes frem. Højt *recall* betyder, at størstedelen af de relevante dokumenter er blevet fundet frem. *Recall* siger derimod ikke noget om, hvor mange irrelevante dokumenter, der er blevet fundet frem sammen med de relevante. Se også *precision*.

Redskab ('tool'). Redskaber kendetegnes ved at overlade den fulde kontrol over løsningen af opgaven til brugeren. Grænserne for et redskabs funktioner er fastlagt af, at intet må foregå bagom ryggen på brugeren. Typeeksemplet på et redskab er hammeren. Redskaber står i modsætning til automatisering, hvor maskiner overtager opgaver, der hidtil er blevet udført af mennesker. Ved automatisering mister/afgiver mennesket kontrollen over og en del af indsigten i løsningen af opgaven.

Relationsdatabaser. Relationsdatabaser kendetegnes ved en enkel model - 2-dimensionelle tabeller hvor hver række er en n-tupel - og ved fysisk og logisk datauafhængighed. Datauafhængigheden betyder, at forespørgselssproget, SQL, er ikke-proceduralt: Forespørgslerne skal beskrive hvilke oplysninger, der skal hentes frem, ikke hvordan de skal findes frem. Et andet væsentligt element i datauafhængigheden er *views*.

Relevans. Relevans er en subjektiv størrelse. Vurderingen af hvilke dokumenter, der er relevante i forhold til en given forespørgsel, kan kun foretages af den bruger, der har stillet forespørgslen. Det skyldes, at relevansen skal vurderes i forhold til den utilstrækkelige viden, forespørgslen udspringer af, ikke i forhold til selve forespørgslen. Specielt har søgesystemets fremfinding af dokumenter intet med en relevansvurdering at gøre. Der kan højst være tale om at forsøge at efterligne menneskers relevansvurderinger ved hjælp af avancerede algoritmer; men søgesystemet finder alene dokumenter frem ud fra sammenligninger af data.
Relevansfeedback ('relevance-feedback'). Relevansfeedback består i en angivelse af, om den forrige forespørgsel fandt for få eller for mange dokumenter frem, eventuelt kombineret med en angivelse af hvilke af de fremfundne dokumenter, der er relevante. Søgesystemet bruger disse oplysninger til automatisk at reformulere forespørgslen, dvs udvide eller indsnævre forespørgslen og muligvis ændre på nogle af søgeordenes vægte. Brugeren skal således ikke selv reformulere forespørgslen, men blot give relevansfeedback ud fra en hurtig, overfladisk vurdering af de dokumenter, den forrige forespørgsel resulterede i.

Skimming ('browsing'). Skimming er en søgeteknik, hvor der fokuseres på den proces, hvor den utilstrækkelige viden formaliseres og repræsenteres. Ved skimming kigger brugeren direkte i dokumenterne, og der er mulighed for at springe fra et dokument til et andet. Ideen er, at brugeren skal 'læse ned over siderne' for at finde noget relevant og have mulighed for at følge en række forskellige associationer, fx springe til et af de dokumenter, der henvises til. Ved skimming organiseres dokumentdatabasen som et netværk: Dokumenterne er netværkets knuder, og associationerne (de kaldes ofte referencerne) udgør forbindelserne mellem knuderne.

Stopliste (‘stop list’). En stopliste er en negativliste; en liste over ord, der ikke har nogen værdi som søgeord i et fuldtekstsøgesystem, og derfor skal filtreres fra under inverteringen. Typiske elementer i en stopliste er ord som 'og', 'en' og 'at'.

System-drevne paradigme ('the system-driven paradigm'). Det system-drevne paradigme var fra begyndelsen af 60'erne til ind i 70'erne det dominerende indenfor forskningen i informationssøgning. Indenfor dette paradigme fokuseres på tekstrepræsentation og søgeteknikker; det er effektiviteten og kvaliteten af selve søgesystemet, der er i centrum. Brugeren ofres mindre interesse, specielt antages det som oftest, at brugerens forespørgsel er identisk med informationsbehovet.

Søgeord. Søgeordene er de ord, som brugeren - sammen med fx de boolske operatorer - bruger til at beskrive sit informationsbehov. Søgeordene er således den del af forespørgslen, der ikke er operatorer.

Søgeteknikker. Søgeteknikker er de teknikker, der bruges til at finde dokumenter frem ud fra de stillede forespørgsler. Søgeteknikker kan deles i to overordnede grupper: Eksakt match teknikkerne og partielt match teknikkerne. I forbindelse med fuldtekstsøgning er boolsk søgning, som er en eksakt match teknik, den mest udbredte. To partielt match teknikker, udvidet boolsk søgning og skimming, er imidlertid også relevante. En god oversigtsartikel over søgeteknikker er (Belkin & Croft 1987).

Tekstrepræsentation. Tekstrepræsentation er den måde dokumenternes tekst er

repræsenteret på i søgesystemet. I dette projekt skelnes mellem to forskellige typer tekstrepræsentation: Fuldtekstrepræsentation og indeksering.

Tesaurus ('thesaurus'). En tesaurus er et kontrolleret glosarium, der er organiseret ud fra den indholdsmæssige sammenhæng mellem de omfattede begreber. I en tesaurus beskrives begreberne ved deres relationer til andre begreber. Typiske relationer er: BT (bredere term), NT (snævrere termer), ST (synonyme termer), RT (relaterede termer), USE (foretrukken term) og SN ('scope note', note om termens form og brug). Et eksempel på en tesaurus er *Computing Reviews Classification System*, den systematik som ACM klassificerer edb-litteraturen efter.

Term-vægtning ('term-weighting', 'vector space' eller 'vector models'). Term-vægtning er en partiel match søgeteknik, hvor hvert dokument beskrives ved en vektor i det vektorrum, der udspændes af alle de potentielle nøgleord. Værdien af hvert af vektorens elementer er det pågældende nøgleords vægt.

Tilstand af utilstrækkelig viden ('anomalous state of knowledge' eller blot 'ASK'). En tilstand af utilstrækkelig viden kan for en persons vedkommende beskrives som *"a certain incompleteness in his picture of the world, an inadequacy in what we might call his "state of readiness" to interact purposefully with the world around him."* (Mackay 1960). Forespørgslerne til et søgesystem udspringer af en sådan tilstand af utilstrækkelig viden.

Udbyder. Udbyderen er den organisation - forlag, avis eller lignende - der markedsfører et informationssøgesystem. Udbyderen er ansvarlig for opdatering og tilføjelse af dokumenter og for systemets øvrige vedligeholdelse.

Udvidet boolsk søgning ('extended boolean retrieval'). Udvidet boolsk søgning er en søgeteknik, der er en kombination af boolsk søgning og term-vægtning. Det er muligt at knytte vægte til såvel ordene i dokumenterne, som ordene i forespørgslerne og de boolske operatorer.

Udvikling over tid. Udvikling over tid betegner det basale forhold, at den kontekst, et søgesystem indgår i, ændrer sig med tiden. Ex: Tidligere var ordet 'husbond' udbredt på landet i betydningen arbejdsgiver, nu har sprogbrugen ændret sig og 'arbejdsgiver' bruges både i byerne og på landet. Et godt søgesystem må støtte udvikling over tid, så ændringer i konteksten kan afspejles i søgesystemet efterhånden, som de opstår.

Viden. Viden er et abstrakt fænomen, der kan opdeles i formaliserbar viden og ikke-formaliserbar viden. Den ikke-formaliserbare viden kaldes ofte tavs viden. Viden opnås gennem erkendelse - lagring af oplysninger er ikke tilstrækkeligt. Viden er derfor noget alene mennesker kan besidde. Se også data og information.

Bilag 2: Stopliste

Dette bilag indeholder en oversigt over de 324 stopord i Edb-Karnovs stopliste.

a
ad
af
al
alene
alle
allerede
alt
altid
anden
andens
andet
andre
andres
at
atter
b
begge
bl
blandt
blev
blevet
blive
bliver
bruge
bruges
burde
bør
c
ca
d
de
del
dels
delvis
delvist
dem
den
denne
dennes
dens
der
deraf
derefter
deres
derfor
deri
dermed
derom
derover

dersom
dertil
derunder
derved
desuden
det
dets
dette
dig
din
disse
disses
du
dog
dvs
e
efter
egentlig
ej
eks
eller
ellers
en
end
endnu
endvidere
enten
er
et
ethvert
evt
eventuel
eventuelle
eventuelt
f
fik
for
foran
fordi
forinden
forvejen
fra
frem
før
få
fået
g
gerne
h
haft

ham
han
har
havde
have
haves
hende
hendes
henholdsvis
her
heraf
herefter
herfor
heri
herimod
hermed
herom
herom
herpå
hertil
herudover
herunder
herved
hidtil
hos
hun
hvad
hvem
hver
hverken
hvert
hvilke
hvilken
hvilket
hvis
hvor
hvoraf
hvordan
hvorefter
hvorfra
hvorhos
hvori
hvorledes
hvormed
hvornår
hvorom
hvorpå
hvortil
hvorunder
hvorved

hvorvidt
i
idet
ifølge
ikke
imens
imellem
imod
incl
ind
inden
indenfor
indtil
ingen
intet
især
iøvrigt
j
jer
jeres
jf
jfr
k
kan
kap
kl
km
kom
kommer
kr
kun
kunne
l
lave
laver
laves
let
lige
lign
lignende
lille
m
man
mange
med
medmindre
meget
mellem
men
mens

mere
mest
meste
mfl
mill
min
mine
mulig
mulige
muligt
mv
må
måtte
n
nedenfor
nogen
noget
nogle
nok
nr
nu
nærmere
nærmest
nødvendig
nødvendige
nødvendigt
o
og
også
om
op
os
over
p
pct
pkt
på
q
r
rimelig
rimelige
rimeligt
s
samme
sammen
samt
samtidig
se
selv
selve

selvom
senere
senest
seneste
ser
set
siden
sig
sige
siger
sin
sine
sit
skal
ske
sker
skulle
snarest
snart
som
stadig
stk
straks
sædvanlig
sædvanlige
sædvanligt
sådan
sådanne
sådant
såfremt
således
sålydende
såvel
t
til
tillige
tilsammen
to
tydelig
tydelige
tydeligt
u
ud
udadtil
ude
udelukkende
uden
umiddelbar
umiddelbare
umiddelbart

under
v
var
ved
vedr
vedrørende
vedrører
vi
videre
vidt
vil
ville
vore
vores
vort
være
været
w
x
y
z
æ
øvrige
øvrigt

Bilag 3: Interviewguide

Dette bilag indeholder den interviewguide, vi brugte i vores indkredsning af juridisk sagsbehandling.

Interviewguide

Indledning:

Båndoptager?
Hvad er din baggrund?
Kan du kort beskrive din placering i organisationen?
Har du erfaringer med Retsinformation eller lignende?

1. Præsentation af vores ideer

Grundlag: Man kan udvikle gode edb-systemer, men ikke intelligente.

Loven er grundstammen i meget juridisk arbejde.
fx noter (i Karnovs Lovsamling), notater eller journaliseringskort kan knyttes til lovens struktur. Søgning skal kunne foregå på samme måde i såvel lovtekst som anden tekst

Søgning i en vilkårlig delmængde af lovteksten og den tilknyttede tekst (ud fra retsområder).

Såvel fuldtekstsøgning som nøgleordsbaseret søgning giver problemer i forbindelse med ordvalget (begreb vs term).
ex: 'husbond' er blevet til 'arbejdsgiver'

Opsamling omkring vores case, Edb-Karnov.
udvikling over tid (notater, dynamisk tesaurus, dynamisk emneregister), fleksibilitet og omfang (ca 400 sider fra Virksomheds-Karnov)

2. Et eksempel på en typisk sag/hvordan du arbejder med loven

Hvilke papirer (dokumenter/gule sedler/papirbunker på skrivebordet/...) giver en sag anledning til?

Hvorfor slår du op i Karnovs Lovsamling...?
-for at sætte dig ind i loven
-for at checke, at du husker loven rigtigt
-for at checke, at det er gældende lov
-på grund af noterne

Foretages opslag (ud fra kendte lov- og/eller paragraf-numre) eller søgning (fx ud fra nøgleord i sagen)?
-i hvilke situationer bruges hvad?

Hvordan foretages søgning?
  -indenfor et retsområde (dvs vha emneregister)
  -vha sagregister
  -vha henvisninger i teksten

Hvilken grundenhed søger du efter/slår op på?
  -love, paragraffer, stk

Hvilken kontekst betragtes/studeres et opslag indenfor?
  -retsområde, lov, paragraf

Bladrer du meget indenfor et opslags kontekst?

Følger du mange af henvisningerne i opslaget/det første opslag?
  -til noter, til andre lovtekster

Hvad bruges opslagene til?
  -sendes til 'kunder', der har spurgt om et eller andet
  -skrives over i en anden tekst
  -er grundlag for en anden tekst, men uciterede
  -andre anvendelser end tekstbehandling?

3. Ønsker/behov/visioner/forbehold ifm edb i sagsbehandlingen

Hvor meget opmærksomhed kan det accepteres at værktøjet kræver?
  -simple værktøjer kræver kun ringe opmærksomhed
  -komplekse værktøjer kræver større opmærksomhed
  -selvfølgelighed/transcendens (‘direct manipulation interface’)

Forestiller du dig blot et søgesystem eller også, at teksterne læses fra skærmen?

Forestiller du dig et fuldtekstsøgesystem og/eller et nøgleordsbaseret system?
  -skal begge dele være til stede?

Vi kunne foreslå følgende funktioner. Hvad synes du om dem?
  -en bogmærke-funktion, så et tidligere opslag kan genfindes/arbejdet kan genoptages dagen efter
  -skimming (følge 'ide')
  -import/eksport, hvorfra/hvortil?

Systemets funktioner kunne støtte:
  -udvikling over tid, fx
    -for hver lovtekst: foregående og efterfølgende lovtekst
    -andre funktioner end dem, vi har tænkt på?
  -begreb kontra term, fx
    -bøjninger, sammensatte ord, varierende stavemåde, ...

Bør systemet også omfatte (principielle) domsafsigelser? Andet?

Ser du store/gode muligheder for brug af edb ifm juridisk sagsbehandling eller begrænsede?

4. Diskussion af vores ideer

Dynamisk emneregister
- -vil en mulighed for en midlertidig gruppering til brug ifm en enkelt sag være relevant?
- -burde emneregistret arbejde med paragraffer?

Dynamisk tesaurus
- -vil dynamikken blive benyttet?
- -ændrer sprogbrugen sig hurtigt?
- -kan det fungere uden central kontrol/udjævning?
- -skal ajourføringen integreres med angivelsen af forespørgslerne for at blive brugt?
- -er der specielle relevante relationer?
- -uden et anerkendt grundtesaurus er det næppe overkommeligt. Findes et sådant? (noget kunne hentes i sagregistret)

Egne notater
- -er der behov for dem (de kan fx bruges til notater om domsafgørelser)?
- -er der behov for flere typer (individuelle, organisationens, ...)?

Vi arbejder med tre søgeniveauer - retsområder, love, paragraffer. Er det tilstrækkeligt/velvalgt?
- -stk
- -midlertidige brugerdefinerede grupper af lovtekster/noter

Læses noterne så ofte, at noterne bør følge lovteksten på skærmen (i hver deres vindue) eller er en opslagsmulighed nok?

Er du interesseret i at se vores prototype (først i maj)?

Bilag 4: Interviewreferater

Dette bilag indeholder de to godkendte referater fra vores interview med:

Dorthe la Cour: jurist i Dansk Arbejdsgiverforening
Per Sjøqvist : advokat og medindehaver af advokatfirmaet Horten & Co

Referat af interview med Dorthe la Cour, den 5. februar 1991

Dorthe la Cour (DC) er jurist og ansat i DA, hvor et af hendes hovedområder er sygedagpenge. Omkring halvdelen af DC's arbejde er at hjælpe og vejlede DA's medlemsvirksomheder og organisationer i konkrete sager. Den anden halvdel består i rådgivning om (videreformidling af) lovændringer. Denne del er for en stor del forebyggende arbejde, da rådgivningen reducerer antallet af konkrete sager, hun involveres i.

1. To typiske sager

DC gav to eksempler på typiske sager. De to sager repræsenterer hver deres ende af, hvordan en sag typisk ser ud. Det første eksempel omfatter blot et enkelt opslag, det andet en søgning, der nærmer sig detektivarbejde. DC vurderer, at en jurists arbejde er mindst lige så meget detektivarbejde, som det er sager, der kan klares direkte ved opslag. Jurister bruger således megen tid på at læse/skimme store mængder lovtekster for at finde relevante oplysninger. Begge eksemplerne er taget fra DC's eget arbejde indenfor den sidste måneds tid.

Eksempel 1: Opslag

> Skal man betale sygeferiepenge til arbejdere, der bliver syge?

Det er faktisk et meget simpelt spørgsmål, idet man simpelthen slår op på ferielovens § 13, og der står hvilke betingelser, der skal være opfyldt osv. I dette tilfælde står svaret direkte i loven; det er ikke nødvendigt at læse noter og lignende. Denne type spørgsmål vil det hverken blive lettere eller sværere at klare med et edb-systems hjælp.

Eksempel 2: Søgning

> Hvad er konsekvenserne af, at man glemmer at fortælle sin arbejdsgiver, at man vil have fædreorlov?

Dette spørgsmål er væsentlig mere kompliceret end det forrige. Man skal for det første vide, at hvis lovteksterne indeholder et svar på dette spørgsmål, så findes det i funktionærloven, lov om ligebehandling af mænd og kvinder eller sygedagpengeloven. For det andet finder man ikke noget om konsekvenserne i disse tre love. Der står noget om, at orloven skal varsles, men ikke noget om konsekvenserne af at glemme det. Det tredje punkt er at søge i egne notater og kollegers notater. Det giver heller ikke noget. Det fjerde punkt er at søge andre steder/på andre områder.

DC fandt svaret i et brev: En virksomhed i en konkret sygedagpengesag havde spurgt dag-pengeudvalget (ankeinstansen for sygedagpengesager) om konsekvenserne. Sygedagpenge-udvalget havde sendt spørgsmålet videre til Arbejdsministeriet. Arbejdsministeriets svar er et brev fra april 1988, og det fandt DC. Dette brev er det eneste sted, hvor svaret på spørgsmålet findes på tryk.

Denne arbejdsproces er mildest talt tung. DC brugte ca en uge på sagen, og det lykkedes kun, fordi hun havde et kontaktnet på deførste ti personer, hun kunne ringe til.

Dette spørgsmål var svært at løse manuelt; men heller ikke med Edb-Karnov ville det have været let: Hvilke(t) ord skulle man søge på? Spørgsmålet kan ikke beskrives ordentligt ved enkeltord; der er behov for en sætning.

Det ville selvfølgelig være nemmere, hvis svaret på et spørgsmål som dette var lagt ind i Edb-Karnov eller et lignende system. Det er imidlertid usandsynligt; DC lægger derfor først og fremmest vægt på, at det ville være rart at kunne gemme svaret, nu hun har fundet det. Så var detektivarbejdet gjort én gang for alle, og svaret derefter lettilgængeligt for alle.

2. Drømmen: Juristens opslagsbog

DC's drøm er at kunne slå op i Edb-Karnov på alle lovtekster, alle Karnovs kommentarer, alle sine egne notater og alt muligt andet: Alt hvad Dagpengeudvalget har forespurgt om, oplysninger om at den og den dom er undervejs, oplysninger om at den og den sag er blevet afgjort osv. Hun understreger imidlertid selv, at det er fuldkommen uladsiggørligt i praksis: Det ville kræve et netværk, hvor alle offentlige instanser skrev ind. Det er imidlertid drømmen for en jurist, da det ville betyde, at alle oplysninger altid kun skulle søges ét sted.

Når DC skal bruge kommentarer til lovteksterne og oplysninger om forarbejderne, slår hun først og fremmest op i Karnovs Lovsamling, i sine egne notater og i sine kollegers notater. DC giver udtryk for, at det er noterne, der er det mest interessante i Karnovs Lovsamling. Lovteksten er som regel let at finde. Det er de forarbejder, fortolkninger osv, der refereres i noterne, som er af værdi, fordi de kan være meget svære at lokalisere. Hvis bare (noterne i) Karnovs Lovsamling og hendes egne og hendes kollegers notater kunne samles ét sted, ville det være en utrolig lettelse.

Juristens opslagbog skal på den ene side give den enkelte jurist en mulighed for at opbevare en masse af det, der ellers står i mapper eller huskes, på en lettilgængelig og varig form. Og på den anden side lette andres adgang til resultaterne af hver enkelt jurists arbejde - en slags vidensbank.

3. Kravene til Edb-Karnov

Jurister står overfor et krav om ikke at overse en eneste relevant lov, lovændring eller lignende. En konsekvens af det er, at mange søgninger i Edb-Karnov eller lignende vil blive meget brede og omfatte meget store retsområder. I de fleste tilfælde giver den ekstra bredde kun irrelevante ting og sager; men der er ikke råd til at miste de få gange, hvor en helt uventet lov, paragraf, henvisning eller lignende dukker op. Edb-Karnov vil derfor blive mødt med tre hovedkrav: Systemet skal være opdateret; det skal være let at få et overblik over, hvad systemet omfatter, og hvad der må søges andre steder; og søgningerne skal være fuldstændige.

Det væsentligste krav: Et opdateret system

En mulighed med edb er at få et system, der er opdateret - Retsinformation er et eksempel. Kravet til et opdateret system er, at ændringer i lovteksten er lagt ind indenfor 24 timer. Man får ikke en jurist til at bruge systemet, hvis ikke opdateringen sker prompte. Hvis systemet ikke er opdateret, kan juristen ikke rådgive med sikkerhed ud fra systemet og skal derfor alligevel til at lave papirarbejde.

Et alternativ til opdatering indenfor 24 timer kunne være opdatering et par gange om ugen kombineret med en tydelig angivelse af tidspunktet for sidste og næste opdatering. Det ville i hvert fald reducere papirarbejdet til perioden fra sidste opdatering frem til nu - hvis man ikke kan leve med 'slippet' eller vente på næste opdatering.

Fuldstændighed og omfang
Jurister søger ofte svar på spørgsmål, der falder indenfor et af de fasttømrede begreber, som fx 'arbejdsskade' eller 'opsigelse'. I disse tilfælde kan man så at sige være sikker på, at en relevant tekst indeholder dette ord. Men det er naturligvis ikke alle sager, der kan karakteriseres ved et af de fasttømrede begreber. Disse sager vil give anledning til omfattende søgninger. Problemet er ikke, at de omfattende søgninger måske tager lang tid; men at der ikke er fuld sikkerhed for, at enhver relevant oplysning bliver fundet frem.

Problemet med fuldstændigheden opstår, fordi der er langt fra mening til formulering. Fuldtekstsøgning alene er utilstrækkeligt, og selvom en tesaurus hjælper, løser den ikke problemet. Tesaurusser arbejder typisk med enkeltord og kan derfor ikke afgøre, at fx 'underslæb' og 'ulovlig omgang med betroede midler' er synonymer. Hvis juristen ikke finder svar på sit spørgsmål, kan hun således aldrig være helt sikker på, at det er fordi, svaret ikke findes i systemet. Hun kan komme ud for ikke at kunne finde svar på et spørgsmål, selvom svaret findes i systemet.

Hvis juristen er overbevist om, at svaret på spørgsmålet ikke findes i systemet, skal hun til at søge andre steder. Det stiller krav om, at det er let at få et overblik over, hvilke kilder og tekster systemet har søgt i.

Søgning kontra læsning
Mange jurister vil sætte pris på, at systemet indeholder præcise referencer til de trykte kilder. Det giver mulighed for udelukkende at bruge systemet til søgning, al læsning kan foretages fra papir. Muligheden for at læse fra papir skal også understøttes ved gode udskriftsfaciliteter. Det giver dels juristen muligheden for at læse fra papir, hvis hun foretrækker det, dels mulighed for at tage en udskrift, når hun sidder med to love, der er knyttet tæt sammen. Det er væsentligt lettere at få overblik over sådanne tekster på papir end ved hjælp af vinduer på skærmen.

Krav til funktionalitet kontra krav til tilgængelighed
DC erkender, at der er en konflikt mellem hvor mange forskellige ting, juristerne gerne vil have edb-systemet kan, og den opmærksomhed, de er villige til at ofre på det. Selv nok så meget brugervenlighed kan ikke gøre brugen af et omfattende edb-system med en kompleks funktionalitet lige så let og selvfølgelig som blyant og papir.

Og glem ikke, at jurister elsker papir.

Papir er alle trygge ved; edb skal derimod bevise sin berettigelse. Årsagen til dette er blandt andet, at jurister påtager sig et anseeligt ansvar, når de rådgiver. De er derfor nødt til at være fuldkomment sikre på, at deres råd hviler på et solidt grundlag. Og papirer, man har læst/skimmet, ér grundigere checket end dokumenter, der er afsøgt for forekomster af en række søgeord.

4. Faciliteter i Edb-Karnov

DC mener, at Edb-Karnov, som vi diskuterer det, er et redskab til fagfolk. Systemet sigter på at gøre juristers arbejde lettere; det vil ikke føre til, at jurister kan erstattes med ikke-jurister hjulpet af Edb-Karnov. Vi er fuldstændigt enige. DC gav udtryk for, at de faciliteter, vi lægger hovedvægten på i Edb-Karnov, er meget relevante.

Egne notater
DC mener, egne notater virkelig vil være guld værd. I Karnovs Lovsamling skriver man i margenen. Det er begrænset, hvad der kan stå der; men det store problem kommer først, når der udkommer en ny udgave af Lovsamlingen: Hvem sidder over overfører sine notater fra de 8000 sider i den gamle Karnov til den nye? I Edb-Karnov kan notaterne overleve opdateringerne.

Der er et stort behov for flere forskellige sæt notater. For DA's vedkommende kunne DC forestille sig: Brevveksling, mellem DA og ministerier, medlemsvirksomheder osv; personlige, som i første omgang kun vedrører den enkelte jurists sager, men hvor kolleger ofte med fordel kunne gå ind og kigge; og notater, dvs redegørelser til direktionen, forretningsudvalget, medlemsorganisationerne og lignende om bestemte juridiske problemer.

Bogmærkefunktion
DC synes, en bogmærkefunktion er en god idé. En simpel bogmærkefunktion er blot et notat med en kort tekst og en dato. DC mener, en mere omfattende funktion også er relevant: De love, der involveres i en søgning, bør afmærkes, og eventuelt bør de søgninger, der udføres i loven, også gemmes. Det vil dels give et overblik over, hvor langt man er kommet i en given søgeproces, dels give mulighed for senere at dokumentere, hvilket grundlag ens anbefaling, svar eller lignende er baseret på.

Dynamisk emneregister
DC lægger størst vægt på muligheden for at vælge de tekster, der skal indgå i søgningen, ud fra emneregistret, dvs ud fra strukturen i lovgivningssystemet. Dynamikken kommenterede hun mest i forbindelse med muligheden for at have en særlig, midlertidig indgang i emneregistret til brug i forbindelse med en enkelt sag. Alle de retsområder, lovtekster osv, der kunne tænkes at være relevante i forbindelse med denne sag, skulle så hæftes på denne indgang. Derefter kunne alle søgninger i forbindelse med sagen udføres indenfor den derved udvalgte del af teksterne i Edb-Karnov.

Love, bekendtgørelser og lignende er ikke sidestillede. DC giver udtryk for, at man ofte vil være interesseret i alle de ting, der er knyttet til en given lov: Bekendtgørelser, vejledninger, andre forarbejder osv. Ofte ved man godt, hvor et eller andet spørgsmål hører hjemme i lovgivningssystemet; men selve lovens tekst er utilstrækkelig. Her er behov for at få alle de oplysning, der er knyttet til fx straffelovens § 13.

Der er imidlertid en risiko forbundet ved at afgrænse søgningerne til udvalgte love: Det er ikke altid, man får alle relevante love med. I nogle tilfælde er det meget svært at afgøre/gætte, hvilke love der kunne være relevante. DC nævner et eksempel: Man har ændret i sygedagpengeloven; men ændringen er foretaget i lov om socialstyrelser. Ændringen er opført sammen med oplysningerne om, hvilke paragraffer der ersattes/ændres med den nye udgave af lov om socialstyrelser. Den rettelse skal man vide er der; det er usandsynligt, at man finder den ved at søge, specielt hvis søgningen afgrænses til udvalgte love.

Dynamisk tesaurus
En tesaurus er uundværlig i forbindelse med fuldtekstsøgning; men den løser ikke alle problemer. Tesaurussen skal udover de juridiske begreber også indeholde noget 'daglig tale', fx de almindeligt anvendte navne for lovene ('sygedagpengeloven' i stedet for 'lov om dagpenge ved sygdom eller fødsel'). Det er urimeligt, at man skal bruge tid på at finde frem til lovens præcise navn, før man kan komme i gang, når man hele tiden har vidst præcis hvilken lov, man var interesseret i.

DC vil foretrække at vedligeholde tesaurussen manuelt som en seperat aktivitet. Hvis ord, der er blevet brugt som fx synonymer i forbindelse med en søgning, automatisk lægges i tesaurussen, frygter hun, at tesaurussen meget hurtigt vil blive stor, rodet og uoverskuelig. En mellemvej mellem manuel og automatisk vedligeholdelse er at lade systemet gemme ordpar, der har været anvendt som synonymer. Juristen kan så med mellemrum hente denne liste frem og se, hvilke ordpar hun mener bør føjes til teaurussen. Den mulighed synes DC er god; hun vil dog gerne have parrene af synonyme ord (og andre oplysninger om hvad og hvor hun søgte) knyttet til den sag, hun brugte dem i.

En liste for hver sag - fx med de brugte søgekommandoer og anvendte synonymer - kan dels bruges som grundlag for ajourføringer af tesaurussen, dels til at gemme synonyme ordpar og lignende, som kun bruges af og til. DC vil helst have en lille tesaurus med de begreber, hun bruger meget tit. Derfor vil hun have fuld kontrol over, hvad der bliver lagt ind i tesaurussen. De ting, hun sjældent bruger, vil hun også gerne have gemt; men det gør ikke noget, at de er lidt sværere at finde frem igen.

Som grundtesaurus foreslår DC det systematiske register i Karnovs Lovsamling. Hun mener, alting i sidste ende bliver hæftet op på strukturen i lovgivningssystemet.

Sagregistret mener hun er en dårligere idé, fordi det ikke ligeså tydeligt afspejler strukturen i lovgivningssystemet. I DA journaliseres meget ud fra lovene, men der journaliseres også ud fra fx virksomheder og brancheforhold. Strukturen i lovene er derfor langtfra den eneste mulighed, når der skal vælges grundtesaurus; valget vil blive meget individuelt.

Historikfacilitet
DC ville gerne have en mulighed for at fremhæve det nye i en ændret udgave af en lov. Den gamle udgave vil man ofte være godt inde i; det er derfor kun ændringerne, man er interesseret i. Man kunne specificere "Ændringer pr den og den dato" og skulle så dels få fremhævet de ting, der blev ændret i lovteksten ved den lejlighed, dels få en oversigt over de lovbekendtgørelser og andet forarbejde, der gik forud for selve ændringen af loven. Det ville give overblik over, hvordan loven havde udviklet/ændret sig.

5. Den mindste tekstenhed og den nødvendige kontekst

Den mindste tekstenhed, man kan nøjes med er et stykke; men paragraffer er at foretrække:

> *En af de første ting, man lærer på universitet [på jurastudiet] er, at man aldrig må læse stykke tre uden at have læst stykke ét og to. Af den grund at stykke ét er hovedreglen og så kommer undtagelserne. Og hvis du ikke har fattet hovedreglen, så forstår du heller ikke undtagelsen, der står i stykke tre.*

Nogle gange er hele lovteksten nødvendig for at kunne vurdere indholdet af en paragraf: Lovtekstens tidligere kapitler har bygget nogle forudsætninger op, der er nødvendige for at forstå indholdet af de senere paragraffer.

6. Retsinformation og brugervenlighed

DC kender Retsinformation, men bruger det nødig. Hun synes, det er ubeskriveligt tungt. Hun sætter brugervenlighed meget højt, og det mener hun, alle jurister gør. DC vil ikke bruge et system, som kræver, at hun skal sidde og spekulere over, hvordan det nu er, systemet skal bruges:

> *Det [Edb-Karnov eller lignende] skal være tilgængeligt. Jeg vil ikke bruge min tid på at sidde og lære et systems mærkelige måde at gøre tingene på. Det skal være utroligt brugervenligt.*

DC mener ikke, Retsinformation opfylder denne betingelse. Hendes væsentligste anker mod Retsinformation er rettet mod de indledende valg af base, valg af hvilke oplysninger om lovteksterne, der skal præsenteres osv. Det er det samlede søgearbejde, der er for stort; det er ikke svartiderne på selve søgningerne, der er problemet.

7. Anvendelsen af de fremfundne tekster

Den lovtekst, der bliver fundet frem, vil typisk skulle overføres til et tekstbehandlingsssystem for at indgå i et svar på et spørgsmål eller et bilag til et internt notat. Ofte bruges citater af lovteksten fremfor henvisninger, dels for at spare læseren for

at skulle slå selve teksten op, dels som dokumentation. Det er ofte en fordel, at det fremgår direkte, at disse retningslinier bygger på den udgave af lovteksten, der har denne ordlyd. Det gør det meget lettere senere at vurdere, hvorvidt retningslinierne stadig gælder, og hvad der eventuelt skal laves om.

Referat af interview med Per Sjøqvist, den 8. februar 1991

Per Sjøqvist (PS) er advokat og medindehaver af advokatfirmaet Horten & Co.

1. To typiske sager
PS giver to eksempler på typiske sager. PS karakteriserer de to sager som henholdsvis en bagudrettet og en fremadrettet. De bagudrettede sager er kendetegnet ved at skulle belyse et bestemt forhold mellem to stridende parter. I belysningen af dette forhold indgår søgning i forskellige love, bekendtgørelserer... De fremadrettede sager er kendetegnet ved at være kreative, da der skal opstilles regler, eksempelvis i en kontrakt til regulering af de fremtidige retsforhold mellem to parter. Eksemplerne er taget fra PS's arbejde, hvor det første er en sag PS arbejder på i øjeblikket.

Eksempel 1: Bagudrettet sagsbehandling (søgning)

> Er der lovhjemmel for, at en bygherre skal betale gebyrrenter på en midlertidig ibrugtagningstilladelse, fra det tidsrum en midlertidig ibrugtagningstilladelse fås, til den endelige erhverves?

Baggrunden for spørgsmålet var, at en bygherre havde fået oplyst af byggestyrelsen, at der ikke var lovhjemmel for gebyrrenter i byggeloven. Bygherren anlagde derfor sag mod Københavns Kommune. PS har set på sagen for Københavns Kommune.

I besvarelsen af spørgsmålet startede PS med at finde byggeloven i Karnovs Lovsamling og læse den. Her var en henvisning til bygningsreglementet. Men der stod ikke noget om gebyrregler disse to steder. Derfor skulle PS finde ud af, hvad der gjaldt før byggeloven af 1976. Før denne gjaldt den københavnske byggelov af 1939. De nuværende gebyrregler er beskrevet her og stammer herfra. Da systemet dengang var anderledes end med det nuværende regelsæt, ønskede PS at spore oprindelsen til dette nærmere.

PS fandt byggelovene fra 1889, 1871 og 1851 frem, men ingen af dem beskriver noget om gebyrrenter. Han søgte derfor litteratur om, hvad praksis havde været. I en bog fra 1928 (der henvises til den i Karnovs Lovsamling) fremgår det, at kommunen i høj grad havde opkrævet gebyrrenter ved midlertidige ibrugtagningstilladelser. For at kunne bevise denne praksis kontaktede PS Stadsarkivet for at se, om kommunens regnskaber viser, at kommunen har oppebåret renteindtægter ved midlertidige ibrugtagningstilladelser.

PS pointerer, at denne sag nok er lidt atypisk ved, at han skulle så langt tilbage i tiden. Men arbejdsgangen i sagen er typisk.

Eksempel 2: Fremadrettet sagsbehandling

> Udarbejdelse af en kontrakt for en totalentreprise.

Kontrakten skal beskrive, hvad totalentreprenøren skal aflevere, hvilken kvalitet det skal have, hvornår, prisen osv. Juristen er i dette arbejde kreativ og skabende i opstillingen reglerne for det fremtidige samarbejde mellem totalentreprenøren og køberen.

Opgaven løses ved at bruge kontorets paradigmaer, tidligere kontrakter og

velgennemtænkte bestemmelser, der tidligere har været brugt i lignende kontrakter. Materialet findes ikke i lovtekster og lignende.

I dette arbejde bruges selve loven meget lidt. I stedet bruges de grundlæggende regler, som PS lærte på jurastudiet, om hvad der gælder mellem to parter. Det er regler om mangler, misligholdelse, forsinkelser. Disse grundregler kan genfindes eksempelvis i købeloven, der således kan danne model for, hvilke principper der gælder på området, og på den måde bruges i forbindelse med kontrakten.

2. Den juridiske metode

PS forklarer, at lovsystemets hierarkiske opbygning med grundloven, lovene, bekendtgørelserne og endelig cirkulærer og vejledninger, hvor der hele tiden sker en trinvis detaljering, afspejles i juristens arbejdsmetode. Den juridiske metode foreskriver, at man i alt juridisk arbejde starter øverst i hierarkiet. Ved en søgning skal man derfor starte med loven og derefter arbejde sig nedad igennem hierarkiet ved hjælp af de henvisninger, der findes undervejs. Alle konkrete problemer, man ikke ved, hvor hører hjemme, skal igennem dette system. Fordelen ved denne arbejdsmetode er, at den afspejler lovsystemets opbygning.

I Karnovs Lovsamling er loven grundpillen, og i noterne kan du finde henvisninger til relevante dokumenter (bekendtgørelser cirkulærer osv). Karnovs Lovsamling understøtter derved den juridiske arbejdsmetode; det er en væsentlig årsag til, at den værdsættes så højt.

Hvis Edb-Karnov med fuldtekstsøgning kan støtte den juridiske metode og gøre juristens arbejde enklere/hurtigere, så er det godt. Men hvis fuldtekstsøgningen medfører for mange tilfældige ordsammenfald og dermed irrelevante dokumenter, vil Edb-Karnov være hindrende for processen og uønsket.

3. Kritik af fuldtekstsøgning

PS kommer med en generel kritik af fuldtekstsøgning på baggrund af hans erfaringer med Retsinformation. PS's indgangsvinkel til søgning er, at han skal være sikker på at få alt det relevante med. Derfor vil han ofte søge bredt.

Valg af søgeord til lovteksten

> *Jeg tror ikke på fuldtekstsøgning i den juridiske verden, fordi de, der laver reglerne, ikke er én person, og derfor ikke skriver de samme ting. De bruger ikke det samme ordforråd. Vi har alle hver vores måde at udtrykke os på, og der er til syvende og sidst en, der har skrevet hvert eneste ord i loven.*

PS kommer med et eksempel: Hvis du vil finde noget om 'reklamation', skal du være klar over, at i købeloven hedder det, at man skal 'gøre sin mangelsindsigelse gældende'. PS mener, at en tesaurus måske delvis kan afhjælpe dette problem.

En flaskehals for erfarne jurister

> *For folk, som har indsigt i det [et retsområde] i forvejen, er fuldtekstsøgning en forbandelse, en vederstyggelighed.*

PS mener, at fuldtekstsøgning har en tendens lave en flaskehals på ens søgning. Man kommer til at bruge for meget tid på søgningen, fordi man skal frasortere mange irrelevante dokumenter.

Hvis man søger på ordet 'byggeri', finder man mange lovtekster. Ud fra lovteksternes navne vil en jurist, der er erfaren, på det pågældende område, hurtigt kunne sortere de irrelevante fra. Men hvis området er nyt for juristen, kan han ikke gøre det og bliver derfor nødt til at læse det hele igennem, da han skal være sikker på at få alt det relevante med.

4. Krav til Edb-Karnov

PS mener, at Edb-Karnov skal være godt, for at jurister vil bruge det; han har svært ved at konkretisere, hvad det præcist dækker, men følgende to punkter er væsentlige for ham.

Overblik

Søgesystemet skal kunne give juristen det samme hurtige overblik, som sagregistret i Karnovs Lovsamling giver. Ved at bladre lidt i registret kan man lynhurtigt finde det, man søger. I arbejdet med søgesystemet må man ikke få fornemmelsen af, at man er på vej ud af en tangent. Det betyder ikke, at man skal finde det søgte i første forsøg, men at man skal kunne finde det uden den store indsats.

> *Hvis det tager mere end 5 minutter at søge ved hjælp af edb, så duer det ikke. Du har simpelthen ikke tid til at vente 5 minutter på at finde det.*

PS nævner flere gange, at for ham er papiret uovertruffent, når det gælder overblik. Når man søger i bøger, har man hele tiden overblikket. Men det har man ikke, hvis man søger ved hjælp af. edb-skærme.

> *Papiret har den evne, at du meget hurtigt får et overblik. På en edb-skærm kan du kun se, hvad du kan se.*

Fuldstændighed

Det er væsentligt, at søgesytemet kan fremfinde alle de relevante dokumenter. Ellers kan PS ikke være sikker på, at han ikke har overset noget. PS mener endvidere, at fremfinding af irrelevante dokumenter vil virke forstyrrende på hans arbejde.

5. Faciliteter i Edb-Karnov

Dynamisk emneregister

PS mener, at det er en god idé, at man selv kan tilføje nye indgange i emneregistret. Jurister er ofte specialiseret indenfor et område, der ikke passer ind i den overordnede opdeling af lovreglerne. Han nævner, at hvis man beskæftiger sig med forureningsret, vil man ofte have brug for regler om forældelse, der ikke har noget med forureningsret at gøre.

PS mener dog, at juristerne allerede i dag laver denne forfining af emneregistret inde i

deres hoved, og at den måske kan være svær at konkretisere. Denne forfining kan de lave på grund af deres uddannelse, paratviden og erfaring.

Egne notater
Det vil være nemmere at løse de fremadrettede sager, hvis sagerne efterhånden kan lægges ind på edb, og det samtidig er muligt at kategorisere dem i fx eneforhandling, entreprise, licensaftale. Herved er det nemmere at finde sagerne frem og genbruge dem.

Tesaurus
I et fuldtekstsøgesystem vil det være nødvendigt med en tesaurus, da der er mange juridiske fagudtryk, der ikke bruges i lovteksterne, fx findes 'reklamation' ikke i lovteksterne, men kaldes 'mangelsindsigelse'. Nytten af en tesaurus stiller PS sig tvivlende overfor, på grund af hans overordnede kritik af fuldtekstsøgesystemer.

Integration i det øvrige arbejde
PS ser ikke det store behov for at kunne klippe noget ud af lovteksten og hente det ind i et tekstbehandlingssystem. Det er sjældent, han citerer paragraffer direkte i sine sager. Endvidere mener PS ikke, tekstbehandling er integreret i juristers arbejde:

> *Jeg kender ikke mange jurister, der sidder og producerer deres eget arbejde ved at indtaste det i tekstbehandling.*

Når han skal sammenskrive nogle dokumenter foretrækker han at gøre det fra papir, da det giver et meget bedre overblik.

6. Indlæggelse af uskreven ret

Erstatningsretten er et eksempel på uskreven ret, og PS mener, at den vil udgøre et problem i et søgesystem, fordi den ikke kan hægtes op på en lov. Da der ikke er nogen skreven ret om erstatningsretten, er det et problem at placere notater, domsafgivelser osv, der handler om erstatningsret. Den uskrevne ret er atypisk for det danske retssystem ved, at man - som i det angelsaksiske retssystem - laver retsregler ud fra konkrete domsafgørelser.

PS nævner, at i de angelsaksiske lande eksisterer der edb-systemer med domsafsigelser. I disse edb-systemer får hver domsafsigelse påhæftet nogle nøgleord, der skal beskrive dets indhold. Disse nøgleord bruges i søgesituationen til at sortere de irrelevante dokumenter fra.

PS mener, at nøgleord er gode i en søgesituation, da de henviser til relevante domsafsigelser. Et lille men ved nøgleord er spørgsmålet om, hvorvidt man skal stole på, at den person, der har valgt nøgleordene, har forstået domsafsigelsen.

Princippet med kun at henvise til de relevante domsafsigelser er Karnovs Lovsamlings styrke. I Edb-Karnov skal man derfor efter PS's mening genfinde dette princip.

7. Den mindste tekstenhed

En søgning skal som hovedregel vise hele loven, bekendtgørelsen eller cirkulæret. Herefter kan man bladre det igennem. I visse tilfælde er loven eller bekendtgørelsen dog

for lang, som fx henholdsvis retsplejeloven med over 1020 paragraffer og bekendtgørelsen om registrering af personkøretøjer med over 150 paragraffer. Det vil i disse tilfælde være nødvendigt at kunne skimme overskrifter, som det i dag er muligt at gøre i Karnovs Lovsamling.

Det vil være problematisk, hvis man i en søgning bliver ført hen til den enkelte paragraf, da man herved ikke finder beslægtede paragraffer i lovteksten. Hvis juristen derimod har et godt kendskab til sit arbejdsområde, skal det være muligt at slå præcis den paragraf, man søger, op.

8. Jurister og edb

PS mener, at jurister ikke kender meget til de muligheder, der ligger i edb. Jurister er forsigtige med at inddrage edb i deres arbejde. PS mener, det tog lang tid før tekstbehandlingssystemer fik en kvalitet, der for alvor gjorde dem anvendelige. Det samme vil være tilfældet med søgesystemer, der ikke i dag har en kvalitet, der gør dem attraktive.

Bilag 5: Afprøvning af Edb-Karnovs svartider

Dette bilag indeholder en beskrivelse af de 30 afprøvningstilfælde, som afprøvningen af Edb-Karnovs svartider omfattede, og de resultater, afprøvningen gav anledning til.

Først beskrives de 30 afprøvningstilfælde. For hvert tilfælde beskrives hvilken tekstmængde der gennemsøges, de anvendte søgeord og de brugte nærhedsoperatorer.

Derefter kommer en skematisk opstilling over resultaterne af afprøvningen. For hvert tilfælde angives svartiden i sekunder og antallet af fundne forekomster.

Med afprøvningen har vi set på svartidernes udvikling både når tekstmængden gradvis øges, og når søgningernes kompleksitet øges. Det fremgår klart af den skematiske opstiling over afprøvningens resultater, at svartiderne ikke er specielt følsomme overfor tekstmængden. De største stigninger er for de svære søgninger, og der er ikke tale om en eksponentiel stigning snarere en logaritmisk. En øgning af søgningernes kompleksitet betyder øgede svartider. De højeste svartider fås således for de mest komplicerede søgninger. Vi finder de opnåede svartider rimelige til demonstrationsbrug. Der er imidlertid gode muligheder for at forbedre dem, da vi kun har gjort en beskeden indsats for at optimere svartiderne. Alle forespørgsler til databasen er opbygget over den samme grundform.

Tilfælde

1. Tekst: lbkg 1985 nr 646, Arbejdsmiljø
   Søgeord: godkendelse(5)[1] og tilsyn(4)[1]
   nærhedsoperator: paragraf

2. Tekst: lbkg 1985 nr 646, Arbejdsmiljø
   Søgeord: godkendelse(5)[1] eller tilsyn(4)[1]
   nærhedsoperator: lov

3. Tekst: lbkg 1985 nr 646, Arbejdsmiljø
   Søgeord: arbejdstilsynet(26)[1] og sikkerhed(29)[1]
   nærhedsoperator: paragraf

4. Tekst: lbkg 1985 nr 646, Arbejdsmiljø
   Søgeord: arbejdstilsynet(26)[1] eller sikkerhed(29)[1]
   nærhedsoperator: lov

5. Tekst: lbkg 1985 nr 646, Arbejdsmiljø
   Søgeord: børn eller aftenarbejde eller arbejde eller barselsorlov eller fogedforretninger eller lægeundersøgelse eller natarbejde
   og
   løn eller ligeløn eller sygdom
   nærhedsoperator: lov

6. Tekst: lbkg 1985 nr 646, Arbejdsmiljø
   Søgeord: funktionærer eller afskedigelse eller anbefaling eller avertering eller bortvisning eller død eller foreningsforhold eller godtgørelse eller graviditet eller kapitalindskud eller konkurrenceklausul eller ligebehandling eller mæglingsmænd eller opsigelse eller organisationsret eller sygdom eller tjenesterejser eller værnepligt
   og
   ferie eller afkald eller arbejde eller feriegodtgørelse eller ferieperiode eller ferietillæg eller ferieår eller forældelse eller kost eller landbrugsbeskæftigede eller logi eller lærlinge eller medhjælperlov eller område eller optjeningsår eller overdragelse eller rekurs eller straf eller sygdom eller tilskadekomst eller tjenestemænd eller værnepligtige
   og
   løn eller ligeløn eller sygdom
   nærhedsoperator: lov

7. Tekst: Retsområde: Arbejdsret (14 lovtekster) + noter
   Søgeord: godkendelse(5) og tilsyn(4)
   nærhedsoperator: paragraf

8. Tekst: Retsområde: Arbejdsret (14 lovtekster) + noter
   Søgeord: godkendelse(5) eller tilsyn(4)
   nærhedsoperator: lov

9. Tekst: Retsområde: Arbejdsret (14 lovtekster) + noter

Søgeord: arbejdstilsynet(26) og sikkerhed(29)
nærhedsoperator: paragraf

10. Tekst: Retsområde: Arbejdsret (14 lovtekster) + noter
Søgeord: arbejdstilsynet(26) eller sikkerhed(29)
nærhedsoperator: lov

11. Tekst: Retsområde: Arbejdsret (14 lovtekster) + noter
    Søgeord: afskedige% (31, 123)[2], overenskomst% (58, 319)[2], berettige% (8, 28)[2]
    nærhedsoperator: lov

12. Tekst: Retsområde: Arbejdsret (14 lovtekster) + noter
    Søgeord: børn eller aftenarbejde eller arbejde eller barselsorlov eller fogedforretninger eller lægeundersøgelse eller natarbejde
    og
    løn eller ligeløn eller sygdom
    nærhedsoperator: lov

13. Tekst: Retsområde: Arbejdsret (14 lovtekster) + noter
    Søgeord: afskedige% (31, 123)[2], overenskomst% (58, 319)[2], berettige% (8, 28)[2]
    nærhedsoperator: paragraf

14. Tekst: Retsområde: Arbejdsret (14 lovtekster) + noter
    Søgeord: funktionærer eller afskedigelse eller anbefaling eller avertering eller bortvisning eller død eller foreningsforhold eller godtgørelse eller graviditet eller kapitalindskud eller konkurrenceklausul eller ligebehandling eller mæglingsmænd eller opsigelse eller organisationsret eller sygdom eller tjenesterejser eller værnepligt
    og
    ferie eller afkald eller arbejde eller feriegodtgørelse eller ferieperiode eller ferietillæg eller ferieår eller forældelse eller kost eller landbrugsbeskæftigede eller logi eller lærlinge eller medhjælperlov eller område eller optjeningsår eller overdragelse eller rekurs eller straf eller sygdom eller tilskadekomst eller tjenestemænd eller værnepligtige
    og
    løn eller ligeløn eller sygdom
    nærhedsoperator: lov

15. Tekst: Retsområde: Beskæftigelse (38 lovtekster) + noter + notater
    Søgeord: godkendelse(5)[1] og tilsyn(4)[1]
    nærhedsoperator: paragraf

16. Tekst: Retsområde: Beskæftigelse (38 lovtekster) + noter + notater
    Søgeord: godkendelse(5)[1] eller tilsyn(4)[1]
    nærhedsoperator: lov

17. Tekst: Retsområde: Beskæftigelse (38 lovtekster) + noter + notater
    Søgeord: arbejdstilsynet(26)[1] og sikkerhed(29)[1]
    nærhedsoperator: paragraf

18. Tekst: Retsområde: Beskæftigelse (38 lovtekster) + noter + notater
    Søgeord: arbejdstilsynet(26)[1] eller sikkerhed(29)[1]
    nærhedsoperator: lov

19. Tekst: Retsområde: Beskæftigelse (38 lovtekster) + noter + notater
    Søgeord: afskedige% (35, 164)[2], overenskomst% (62, 375)[2], berettige% (10, 91)[2]
    nærhedsoperator: lov

20. Tekst: Retsområde: Beskæftigelse (38 lovtekster) + noter + notater
    Søgeord: børn eller aftenarbejde eller arbejde eller barselsorlov eller fogedforretninger eller lægeundersøgelse eller natarbejde
    og
    løn eller ligeløn eller sygdom
    nærhedsoperator: lov

21. Tekst: Retsområde: Beskæftigelse (38 lovtekster) + noter + notater
    Søgeord: afskedige% (35, 164)[2], overenskomst% (62, 375)[2], berettige% (10, 91)[2]
    nærhedsoperator: paragraf

22. Tekst: Retsområde: Beskæftigelse (38 lovtekster) + noter + notater
    Søgeord: funktionærer eller afskedigelse eller anbefaling eller avertering eller bortvisning eller død eller foreningsforhold eller godtgørelse eller graviditet eller kapitalindskud eller konkurrenceklausul eller ligebehandling eller mæglingsmænd eller opsigelse eller organisationsret eller sygdom eller tjenesterejser eller værnepligt
    og
    ferie eller afkald eller arbejde eller feriegodtgørelse eller ferieperiode eller ferietillæg eller ferieår eller forældelse eller kost eller landbrugsbeskæftigede eller logi eller lærlinge eller medhjælperlov eller område eller optjeningsår eller overdragelse eller rekurs eller straf eller sygdom eller tilskadekomst eller tjenestemænd eller værnepligtige
    og
    løn eller ligeløn eller sygdom
    nærhedsoperator: lov

23. Tekst: Alle lovtekster + noter + notater
    Søgeord: godkendelse(5)[1] og tilsyn(4)[1]
    nærhedsoperator: paragraf

24. Tekst: Alle lovtekster + noter + notater
    Søgeord: godkendelse(5)[1] eller tilsyn(4)[1]
    nærhedsoperator: lov

25. Tekst: Alle lovtekster + noter + notater
    Søgeord: arbejdstilsynet(26)[1] og sikkerhed(29)[1]
    nærhedsoperator: paragraf

26. Tekst: Alle lovtekster + noter + notater
    Søgeord: arbejdstilsynet(26)[1] eller sikkerhed(29)[1]
    nærhedsoperator: lov

27. Tekst: Alle lovtekster + noter + notater
    Søgeord: afskedige% (36, 186)[2], overenskomst% (66, 425)[2], berettige% (12, 189)[2]
    nærhedsoperator: lov

28. Tekst: Alle lovtekster + noter + notater
Søgeord: børn eller aftenarbejde eller arbejde eller barselsorlov eller fogedforretninger eller lægeundersøgelse eller natarbejde
og
løn eller ligeløn eller sygdom
nærhedsoperator: lov

29. Tekst: Alle lovtekster + noter + notater
Søgeord: afskedige% (36, 186)[2], overenskomst% (66, 425)[2], berettige% (12, 189)[2]
nærhedsoperator: paragraf

30. Tekst: Alle lovtekster + noter + notater
Søgeord: funktionærer eller afskedigelse eller anbefaling eller avertering eller bortvisning eller død eller foreningsforhold eller godtgørelse eller graviditet eller kapitalindskud eller konkurrenceklausul eller ligebehandling eller mæglingsmænd eller opsigelse eller organisationsret eller sygdom eller tjenesterejser eller værnepligt
og
ferie eller afkald eller arbejde eller feriegodtgørelse eller ferieperiode eller ferietillæg eller ferieår eller forældelse eller kost eller landbrugsbeskæftigede eller logi eller lærlinge eller medhjælperlov eller område eller optjeningsår eller overdragelse eller rekurs eller straf eller sygdom eller tilskadekomst eller tjenestemænd eller værnepligtige
og
løn eller ligeløn eller sygdom
nærhedsoperator: lov

Noter

1. Antal gange ordet forekommer i Arbejdsmiljøloven.

2. Det første tal er antallet af forskellige ord, som matcher søgeordet. Der er mere end ét, da søgeordet indeholder en joker-operator. Det andet tal er det samlede andtal forekomster for alle de ord, der matcher søgeordet.

Afprøvningsresultater

	Forsøg 1: 116 sider		Forsøg 2: 218 sider		Forsøg 3: 290 sider		Forsøg 4: 407 sider		Forsøg 5: 407 sider	
Testtilfælde	Tid	Antal	Tid	Antal	Tid	Antal	Tid	Antal	Tid	Antal
1.	1	0	1	0	1	0	1	0	1	0
2.	1	9	1	9	1	9	1	9	1	9
3.	1	6	1	6	1	6	1	6	1	6
4.	1	49	1	49	1	49	1	49	1	49
5.	1	1	1	1	1	1	1	1	1	1
6.	2	2	2	2	2	2	2	2	2	2
7.	2	0	2	0	2	0	2	0	2	0
8.	1	5	1	5	3	5	3	5	3	5
9.	3	1	3	1	3	1	3	1	3	1
10.	3	1	3	1	3	1	3	1	3	1
11.	7	4	7	4	7	4	7	4	7	4
12.	6	12	7	12	7	12	8	12	8	12
13.	7	2	7	2	7	2	7	2	7	2
14.	20	14	24	14	24	14	26	14	26	14
15.	2	2	2	2	2	2	4	2	4	2
16.	2	23	2	23	2	23	2	23	2	23
17.	5	9	5	9	5	9	5	9	5	9
18.	5	30	4	30	3	30	4	30	4	30
19.	7	4	7	4	7	4	7	4	7	4
20.	9	23	9	23	9	23	9	23	9	23
21.	7	2	7	2	7	2	9	2	9	2
22.	28	25	28	25	28	25	30	25	29	25
23.	2	2	2	3	4	9	4	10	5	10
24.	2	23	2	42	4	59	4	65	4	65
25.	4	9	4	9	4	9	4	9	4	9
26.	5	30	4	38	4	43	5	54	4	54
27.	8	4	8	4	8	4	8	4	8	4
28.	8	23	9	33	9	33	10	35	10	35
29.	7	2	9	2	9	2	8	2	8	2
30.	27	25	28	37	30	37	34	41	32	41

Bilag 6: Oversigt over Edb-Karnovs pladsforbrug

Dette bilag indeholder en oversigt de forskellige tabellers og indeks's pladsforbrug i Edb-Karnov i forbindelse med indlæggelse af større og større tekstmængde. Alle tallene er angivet i Kb.

Af oversigten fremgår, at det er lovteksterne med tilhørende noter, der bruger den altovervejende del af pladsen. Lovteksternes og noternes pladsforbrug er fordelt på teksttabellerne, søgeordstabellerne og disse tabellers indeks.

Edb-Karnovs database fylder knap 19 Mb, når de 407 sider fra Virksomheds-Karnov er lagt ind. I almindelig tekst, dvs som ASCII-fil, fylder de samme 407 sider ca 4 Mb. I Edb-Karnov er der således et ekstra pladsforbrug på ca 400% i forhold til det, selve teksten fylder. Ved brug af inverterede filer er stigningen i pladsforbrug typisk 50-300% i forhold til den oprindelige tekstmængde. Pladsforbruget er således større ved brug af relationsdatabaser; men brugen af en relationsdatabase har ikke givet anledning til en drastisk stigning i pladsforbruget.

Pladsforbruget i Edb-Karnov

Tabel/indeks	Indlagt 116 sider	Indlagt 218 sider	Indlagt 290 sider	Indlagt 407 sider
LovOplysninger, data	30	30	30	30
LovOplysninger, indeks	5	30	30	30
LovTekst, data	755	1.530	1.955	2.455
LovTekst, indeks	30	30	30	30
LovSøgeord, data	855	-	2.205	2.780
LovSøgeord, indeks	730	1.455	1.880	2.380
NoteTekst, data	1.155	1.780	2.430	3.280
NoteTekst, indeks	30	55	80	105
NoteSøgeord, data	1.280	2.255	3.180	4.105
NoteSøgeord, indeks	1.080	1.905	2.605	3.505
NotatTekst, data	5	5	5	5
NotatTekst, indeks	5	5	5	5
NotatSøgeord, data	5	5	5	5
NotatSøgeord, indeks	5	5	5	5
EmneRegister, data	5	5	5	5
Emneregister, indeks	5	5	5	5
EmneRegOpdeling, data	5	5	5	5
EmneRegOpdeling, indeks	5	5	5	5
EmneRegisterLov, data	5	5	5	5
EmneRegisterLov, indeks	5	5	5	5
TesaurusTermer, data	30	30	30	30
TesaurusTermer, indeks	30	30	30	30
TesaurusHierarki, data	30	30	30	30
TesaurusHierarki, indeks	30	30	30	30
Synonymer, data	5	5	5	5
Synonymer, indeks	5	5	5	5
TermReference, data	30	30	30	30
TermReference, indeks	5	5	5	5
SøgLovid, data	5	5	5	5
SøgLovid, indeks	5	5	5	5
StopListe, data	30	30	30	30
StopListe, indeks	5	5	5	5
Pladsforbrug i alt	**6.210**	**11.035**	**14.685**	**18.960**

Alle tal angiver pladsforbruget i Kb.