

Generisk intelligente søgesystemer

Henrik Bulskov

Datalogivejleder: Troels Andreasen

Kommunikationsvejleder: Henning Ørum

Kombinations speciale

Datalogi og kommunikation

Roskilde Universitetscenter

Oktober 2001

Forord

Dette er en udgave af specialet *Generisk intelligente søgesystemer* uden kildekode til den prototype der udvikles. Referencer til implementeringen (appendiks) kan derfor være misvisende. Eventuelle spørgsmål til implementeringen kan rettes til bulskov@ruc.dk.

Dette projekt er et kombinationsspeciale i datalogi og kommunikation. Det overordnede mål med specialet har været at undersøge, om et fleksibelt intelligent søgesystem kan understøtte bibliotekarers ønsker og behov i forbindelse med bibliografisk søgning. Projektet beskriver den første fase i udvikling af et søgesystem, fra den første kontakt til målgruppen frem til en afprøvning af første prototype. Projektet beskæftiger sig med forskning i søgesystemer og muligheden for at benytte kommunikative videnskabelige metoder for at understøtte denne forskning. Prototypens fleksibilitet i forbindelse med evaluering, opnås gennem udvidelse af forespørgsler. Ved at inddrage viden, under evalueringen af forespørgslen, kan systemet tilføje information som er relateret til, men ikke direkte dækket af forespørgslen. Findes der for eksempel viden om, at *bil* og *automobil* er synonyme, kan en forespørgsel på *bil* udvides til ”*bil* eller *automobil*”. I projektets begyndelse gennemførtes en interviewundersøgelse, som viste, at der var en vis modstand og skepsis mod intelligente søgesystemer. Efter den afsluttende afprøvning af prototypen var holdningen til de anvendte metoder imidlertid ændret. Informanterne konstaterede at metoderne indeholder muligheder, som det klart vil være interessant at udforske yderligere.

I forbindelse med udarbejdelse af specialet vil jeg gerne takke:

- Mine vejledere på henholdsvis kommunikation og datalogi, Henning Ørum og Troels Andreasen, for en tålmodig og engageret vejledning og konstruktiv kritik.
- Allan Forsberg for levering af datagrundlag samt råd og vejledning.
- Birthe Randrup for korrekturlæsning.

- Else Hansen og Kirsten Styltsvig for transkription af interviews.

Roskilde, september 2001

Indhold

1	Indledning	1
1.1	Den kommunikative vinkel	2
1.2	Den datalogiske vinkel	3
1.3	Problemafgrænsning	6
1.4	Problemformulering	8
2	Metode	9
2.1	Kommunikativ forforståelse	9
2.2	Systemudvikling	10
2.3	Søgesystemet	12
2.3.1	Programmeringstekniske valg	13
3	Interviewundersøgelse	15
3.1	Interview	15
3.1.1	Repræsentativitet	15
3.1.2	Gyldighed og pålidelighed	17
3.1.3	Interviewguide	18
3.2	Interviewanalyse	19
3.2.1	Den nuværende brug	20
3.2.2	Det fremtidige søgesystem	22
3.2.3	Opsamling	22
4	Design af prototype	24
4.1	Datagrundlag	25
4.2	Repræsentation	26
4.2.1	Objekter	28
4.2.2	Metainformation	29
4.2.3	Visningsformat	31
4.2.4	Objekter, metainformation og visningsformat	33
4.2.5	Indeksring	34
4.2.6	Viden	35
4.3	Brugergrænseflade	37
4.3.1	Tegnbaserede grænseflader	38

4.3.2	Web-baserede grænseflader	39
4.3.3	Prototypens grænseflade	40
4.3.4	Associationsnavigator	41
4.4	Associationsnet	43
4.5	Forespørgselsevaluering	46
4.5.1	Fuzzy-mængder	47
4.5.2	Order Weighted Average aggregering	48
4.5.3	Forespørgselsprog	49
5	Implementering af prototype	51
5.1	Database	51
5.2	Databehandling	52
5.3	Forespørgselsevaluering	55
6	Diskussion	59
6.1	Afprøvning	59
6.1.1	Eksempler	60
6.1.2	Informanternes afprøvning af prototypen	62
6.1.3	Opsummering	63
6.2	Design og implementation	64
6.3	Interviewundersøgelse	66
6.4	Konklusion	67
	Litteratur	69

Kapitel 1

Indledning

Motivationen for dette projekt udspringer af mit arbejde i forskningslaboratoriet Intelligent Systems Laboratory (ISL) på RUC og af mine tidligere projekter [Bulskov, 1998], [Brix *et al.*, 1998]. ISL's primære forskningsområde er intelligente søgesystemer, og gennem arbejdet med dette har jeg været, og er, involveret i en række forskellige forskningsprojekter, der alle omhandler søgning i elektroniske databaser. Projekternes fællesnævner er ønsket om at kunne forbedre søgesystemers evaluering af forespørgsler og dermed de svar, der præsenteres for brugerne. Projekterne har involveret eksterne samarbejdspartnere, både private og offentlige [Andreasen and Bulskov, 1998], [Andreasen, 1998], [Forsberg *et al.*, 1999], [Andreasen *et al.*, 2000], og har været finansieret af offentlige- og private midler. Formålet har derfor både haft karakter af grundforskning og specifik problemløsning.

Mit mål for dette projekt har været et kombinationsspeciale i kommunikation og datalogi. Jeg ville derfor flette discipliner fra disse to fag sammen og undersøge, om det var muligt at skabe et fundament for den datalogiske forskning, der kunne lede til en bedre forståelse af målgruppen og en bedre afprøvning af de systemer, der udvikles. Forskning i intelligente søgesystemer vil ofte indeholde kommunikative problemstillinger, fordi hele formålet med et søgesystem er, at der på baggrund af et input fremkommer et output fra systemet. Dette vil, uanset hvad eller hvem der definerer input og forholder sig til output, være en kommunikativ proces. Datalogi og kommunikation har en del overlappende områder, hvor faggrænserne er meget udflydende. Et eksempel herpå er hele det område, der kaldes Human-Computer Interaction. Det er dog vanskeligere at kombinere de to fag, når udgangspunktet er specifikke elementer fra det ene eller det andet fag. I dette tilfælde, med intelligente søgesystemer, er det oplagt at brugergrænseflader og funktionalitet er et fælles anliggende, mens de teoretiske og tekniske elementer i selve søgesystemet i højere grad er datalogi. På samme måde vil en detaljeret målgruppeanalyse og modeller til analyse af interview i højere grad være

kommunikation. Det var min overbevisning, at inddragelse af kommunikative videnskabelige metoder ville kunne understøtte forskning i intelligente søgesystemer, således at der dels kunne skabes et bedre overblik over brugernes ønsker og behov, og dels kunne etableres en bedre afprøvning gennem denne indsigt i målgruppen.

1.1 Den kommunikative vinkel

I den del af forskningen i ISL, der har involveret private samarbejdspartnere, har målet ofte været en forbedring eller en funktionel udvidelse af eksisterende systemer. Det eksisterende system har haft en eller flere målgrupper, og forskningen har haft til formål at udvikle systemer, der kunne understøtte en eller flere af disse.

Selvom det eksisterende system havde forskellige målgrupper, indgik disse ikke, på videnskabelig vis, som selvstændig del af forskningen. De blev ikke inddraget i forskningsprocessen, og deres behov blev vurderet af deltagerne i forskningen på baggrund af deres kendskab til målgruppens vaner, behov og ønsker. Dette er der i sig selv ikke noget forkert i, hvis formålet med forskningen er at undersøge forskellige teorier og teknikkers muligheder og holdbarhed. Men i det omfang, at forskningsprojekterne skal testes og evalueres af andre end deltagerne i forskningsprojektet, kan der opstå vanskeligheder med udvælgelse af respondenter. Et eksempel kan være udformningen af en grænseflade til de systemer, der udvikles, som i høj grad kræver kendskab til målgruppen, fordi der vil være meget stor forskel på, hvad de erfarne og de uerfarne brugere har behov for. Passer grænsefladen ikke nogenlunde overens med brugerens ønsker og behov, kan det meget let ende med at være en test af grænsefladen og ikke det underliggende system, fordi brugeren har vanskeligt ved at abstrahere fra de grænseflademæssige problemstillinger.

Den type af søgesystemer, der er i fokus i dette projekt, er systemer, der på baggrund af en tekstuel beskrivelse kan finde elektronisk lagrede tekstuelle objekter. Det primære mål for disse søgesystemer er hurtigt og effektivt at finde den delmængde af de repræsenterede objekter, der ønskes af brugeren. Dette er ikke en simpel proces, idet den kommunikation, der skal foregå mellem søgesystemet og brugeren, er begrænset af den grænseflade søgesystemet stiller til rådighed.

Det første, brugeren skal foretage sig, er at omsætte den viden eller mangel på samme, han eller hun har om det, der ønskes fundet, til det format, som søgesystemet kan forstå. Denne proces er ofte ensbetydende med, at der skal ske en transformation af information, fra brugerens abstrakte og komplekse tanker, til et logisk og stringent forespørgselsprog. Her vil der utvivlsomt

være information, der går tabt eller bliver ændret til noget, der ikke mere beskriver det udgangspunkt, brugeren havde. Dette kan ske alene ved at skulle transformere fra det tænkte sprog til det skrevne. Det betyder, at der kan være meget forskel på det, en bruger tænker om det, der ønskes fundet, og den beskrivelse, der videregives til søgesystemet gennem grænsefladen. Dette er første led i en lang række af mulige misfortolkninger, der kan resultere i, at det returnerede svar ikke stemmer overens med den forventning, brugeren havde.

Selvom søgesystemer kunne fortolke og forstå natursprog, ville dette ikke i sig selv løse ovenstående problemstilling, fordi fortolkning og forståelse forudsætter et fælles kontekstuel grundlag. Er dette kontekstuelle grundlag ikke tilstede, vil der, også i mellem menneskelig kommunikation, opstå misforståelser. Det er altså vigtigt, at søgesystemer hjælper brugeren til at fastlægge det kontekstuelle grundlag. Søger en bruger for eksempel efter bøger om *mus*, er der meget forskel på, om konteksten er *kæledyr* eller *computerudstyr*. Naturligvis kan systemet blot levere alle de objekter, der omhandler *mus*, uanset kontekst, men forventer brugeren noget i konteksten *kæledyr*, og de første mange objekter er fra konteksten *computerudstyr*, vil svaret ikke umiddelbart opfylde brugerens forventninger.

Overordnet skal den kommunikative vinkel belyse, hvordan inddragelse af kommunikative metoder i et projekt, der belyser snævre datalogiske problemstillinger, kan tilføre viden og indsigt, der direkte kan benyttes i de beslutninger, der skal træffes omkring den datalogiske del. Den kommunikative indsigt og forståelse, skal danne den overordnede ramme, hvori datalogien udvikles, og dermed lede til en bedre afprøvning gennem indsigt i målgruppen. Samtidigt skal den kommunikative forståelse afstikke rammerne for den kommunikation, der skal etableres mellem søgesystemet og brugeren, og dermed være med til at reducere mulighederne for, at der opstår misforståelser, der medfører dårlige svar.

1.2 Den datalogiske vinkel

Grundlaget for søgesystemer i denne kontekst er tekstuelle objekter. Der kan skelnes mellem objekter på to niveauer: indhold og struktur. Den indholdsmæssige del omhandler den semantiske forståelse af objekterne, for eksempel hvilket domæne objektet tilhører, eller forståelse af det emne, indholdet beskriver. Den strukturelle del beskriver, hvordan objektet rent fysisk fremtræder. Nogle objekter vil have en stringent struktur, der definerer, hvilke dele det består af, for eksempel en titel, emneord, ISBN nummer, et cetera, mens andre har en mere tilfældig struktur, hvor det vil være sværere at udlede de enkelte dele.

I dette projekt er det primært problemstillinger omkring det strukturelle niveau, der behandles. Målet er at designe et søgesystem, der er fleksibelt i forhold til objekternes strukturelle formatering, således at det kan behandle forskellige objekter med forskellig struktur dynamisk, primært ved at definere et format, alle objekter skal overholde, og ved at tilknytte strukturelle beskrivelser til de forskellige objekttyper.

Målet med forskning i søgesystemer er at undersøge, hvordan søgesystemets funktionalitet og effektivitet kan forbedres. Forholdet mellem funktionalitet og effektivitet er ofte to modsætninger, der trækker i hver sin retning. Vil man gerne tilføje funktionalitet, der fortolker forespørgslen på en smartere måde og dermed fremkommer med kvalitativt bedre svar, vil det som regel betyde, at systemet skal bruge mere tid til evalueringen af forespørgslerne. Vil man forbedre systemets svartider, må designet optimeres, og antallet af processer i evalueringen reduceres. Det er min overbevisning, at et kvalitativt bedre svar i de fleste tilfælde er at foretrække fremfor for et hurtigt svar. Men naturligvis er der en grænse for, hvor lang tid systemet må bruge på evalueringen af forespørgsler. Denne vægtning mellem kvalitative bedre svar og hurtige svar er en af de spændende problemstillinger i forskning i søgesystemer.

Den forskning jeg har været involveret i gennem mit arbejde i ISL, har beskæftiget sig med et begreb, der kaldes *fleksible søgesystemer*. Hvor det fleksible kan handle om at konstruere søgesystemer, der er mere tolerante overfor, hvordan brugerne indtaster forespørgslen og hvor stringent det, der udtrykkes, skal fortolkes. Dette kunne være ved at acceptere forespørgsler udtrykt i natursprog eller ved at eliminere brugen af booleske operatører, for at give et par eksempler på, hvordan en større tolerance kan opnås. Den mindre stringente fortolkning kunne være, at systemet benytter den viden, det indeholder omkring datagrundlaget, til at hjælpe brugerne, når forespørgslen resulterer i et svar med for få eller for mange objekter, for eksempel ved at give forskellige løsningsforslag i stedet for.

Fleksible søgesystemer bliver ofte kaldt *intelligente søgesystemer*, fordi det at bringe viden i anvendelse i et søgesystem simulerer menneskelige problemløsning. Det er dog nødvendigt at forholde sig til begrebet intelligens, fordi der er mange forskellige holdninger til, hvordan man overhovedet definere intelligens, og ikke mindst fordi intelligens, når det benyttes i forbindelse med computerteknologi, giver anledning til endnu mere uenighed¹.

¹En element i uenigheden er, at det er umuligt at definere kunstig intelligens, når man ikke kan definere den "ægte" intelligens. Et andet er, om man overhovedet kan tale om intelligens i forbindelse med computerteknologi. Det er en diskussion, der ligger udenfor dette projekts rammer, og som derfor ikke vil blive belyst yderligere.

Intelligens er i denne kontekst en speciel form for kunstig intelligens, hvor det handler om at simulere intelligente processer. Målet er ikke at forholde sig til, eller eftergøre, hvordan mennesker ved hjælp af intelligens løser et problem, men derimod at forsøge at tilnærme sig det resultat, der fremkommer fra den menneskelige problemløsning. Dette kan lettest illustreres ved at give et eksempel. En person spørger en bibliotekar om bøger, der omhandler begrebet *børn*. Bibliotekarens første reaktion vil formentlig være at bede om en uddybning, fordi begrebet *børn* er alt for bredt til at give et konkret svar. Det intelligente søgesystem skal altså ikke gøre det samme ved at forsøge at eftergøre de processer, der får bibliotekaren til at bede om en uddybning, men derimod ved at benytte den viden, systemet indeholder. For eksempel at der er mange objekter, der handler om *børn*, og derfor skal søgesystemet resolve, at det skal inddrage de begreber, *børn* ofte optræder sammen med. Svaret på en søgning efter begrebet *børn* i et intelligent søgesystem, skal med andre ord kunne resultere i en uddybning af, hvad *børn* optræder sammen med, for eksempel *leg, sygdom, forældre, skole*, et cetera.

Det intelligente søgesystem skaber simuleringen af intelligens ved at trække på den viden, der er tilknyttet systemet. I ovenstående eksempel, er det bibliotekarens viden og indsigt, der er forudsætningen for at kunne konkludere, at det er nødvendigt med en uddybning af emnet *børn*, for at kunne give et konkret svar. Det er også viden, der gør det intelligente søgesystem i stand til at resolve anderledes end det ikke intelligente søgesystem. Kvaliteten af den viden, der tilknyttes det intelligente søgesystem, bestemmer, hvor godt simuleringen af de intelligente processer opfattes. Hvis de begreber, der er relateret til et begreb er dårlige og uforståelige for brugeren og ikke afspejler bibliotekarens viden, vil den tilsigtede effekt ikke kunne opnås. Den vil tværtimod blive opfattet som irriterende og misvisende. Målet er derfor, at den tilknyttede viden skal være af en høj kvalitet, og at den skal benyttes varsomt.

Den viden, der tilknyttes søgesystemer kan blandt andet være deklarativ eller relationel. Den deklarative viden er baseret på regler, der kan benyttes til at udlede viden fra bestemte sammenhænge, for eksempel de regler, der benyttes til forståelse af syntaks eller semantisk fortolkning i natursprogsanalyse. Den relationelle viden er baseret på relationelle sammenhænge, for eksempel synonymi og ontologi. I dette projekt er det inddragelse af relationel viden, jeg vil beskæftige mig med², og hvordan denne viden kan benyttes i søgesystemet til at fremkomme med kvalitativt bedre svar.

Et problem ved konstruktionen af søgesystemer, der inddrager viden, er, at de ofte tilpasses bestemte typer viden og derfor ikke kan tilknyttes an-

²Jeg vil herefter benytte begrebet viden i betydningen relationel viden.

dre typer af viden uden at skulle rekonstrueres. For eksempel vil et system, der kan benytte synonymi, ikke uden videre kunne tilknyttes holonymi, og benytte denne nye viden. Det er derfor min hensigt at konstruere et generisk søgesystem, hvor inddragelse af viden er dynamisk, således at systemet kan anvendes med forskellige typer af viden, uden at skulle reprogrammeres.

1.3 Problemafgrænsning

Den kommunikative vinkel lægger op til et forløb, hvor en målgruppe undersøges og der udledes viden om dennes ønsker og behov i forhold til søgning, som så danner grundlaget for en række datalogiske valg. Afslutningsvis skal disse valg evalueres i den valgte målgruppe for at afdække, hvorvidt valgene formår at tilføje funktionalitet, der kvalitativt forbedrer søgesystemet.

Udviklingen af et søgesystem skal i denne kontekst ansues fra to vinkler: de ønsker og behov, der klarlægges gennem undersøgelse af målgruppen, og ønsket om at afprøve forskellige datalogiske metoder og teorier. Inden for de seneste år har der inden for systemudvikling været en tendens til, at tage udgangspunkt i at brugeren altid har ret. Dette kan naturligvis være et meget fornuftigt udgangspunkt, hvis man skal sælge en vare eller et budskab. Men hvis målet er at undersøge og udforske nye teknologier, kan det være problematisk, fordi der ofte er en tendens til, at brugerne har en iboende modstand mod forandringer [Beyer and Holtzblatt, 1998]. Målet i dette projekt bliver derfor at finde et kompromis mellem brugernes ønsker og ønsket om at introducere nye teknologier.

Dette projekt skal ses som første fase i et udviklingsforløb, der skal føre til udviklingen af et søgesystem. Projektet skal belyse den første fase i denne udvikling og behandle forløbet fra den første kontakt til målgruppen frem til implementering af en prototype og afprøvning af denne. I forlængelse af projektet skal resultatet af den første afprøvning bruges til at videreudvikle prototypen. Denne proces skal fortsætte indtil prototypen kan danne grundlag for implementering af den første version af søgesystemet.

Valget af datagrundlag er essentielt i forskningsprojekter, der beskæftiger sig med søgning. Specielt er et lille datagrundlag ofte skyld i, at de udviklede metoder og teorier ikke kan generaliseres og dermed ikke kan realiseres på andre datamængder. For eksempel kan en metode, der på et lille datagrundlag er hurtig og effektiv, på en større datamængde vise sig at være ineffektiv og langsom. For at undgå denne problemstilling, og fordi jeg gennem flere år har været tilknyttet et forskningsprojekt, der har arbejdet med bibliografisk

søgning [Andreasen, 1998], valgte jeg, at datagrundlaget for dette projekt skulle være databasen DanBib. DanBib administreres af Dansk Biblioteks-Center (DBC) og indeholder blandt andet bibliografiske informationer fra samtlige folke- og forskningsbiblioteker i Danmark. DanBib indeholder cirka 14 millioner bibliografiske referenceposter, der beskriver bibliografiske objekter³.

Der vil naturligvis være en sammenhæng mellem det valgte datagrundlag og den målgruppe, søgesystemet henvender sig til. Med DanBib som grundlag skulle målgruppen findes blandt personer, der har behov for søgning efter bibliografiske objekter. Denne målgruppe repræsenterer brugere, hvis forhold til søgning fordeler sig i hele spekteret fra novicen til superbrugeren. Det var min vurdering, at gruppen af superbrugere i højere grad ville være i stand til at forholde sig til intelligente søgesystemer på et abstrakt niveau, qua deres erfaring og daglige brug af mange forskellige søgesystemer. Som tidligere nævnt vil udviklingsforløbet kræve, at brugerne konsulteres flere gange, hvilket giver mulighed for at ændre eller udvide målgruppen senere i processen, hvis dette findes nødvendigt. Jeg valgte derfor at fokusere på superbrugerne i denne første fase, og specielt det segment af superbrugere, der til dagligt professionelt beskæftiger sig med bibliografiske søgninger, bibliotekarer.

I forbindelse med projektet *Fleksibel søgning i DanBib* [Andreasen and Bulskov, 1998] var datagrundlaget, der blev benyttet, begrænset til de objekter i DanBib der havde en speciel type emneord tilknyttet. Dette gav en lang række problemer i forbindelse med afprøvning af den prototype, der blev udviklet. Det var ikke muligt at lave en realistisk sammenligning med andre systemer, der benyttede hele DanBib som datagrundlag, fordi svarene ikke var sammenlignelige. Det var heller ikke muligt, at bede brugerne om at løse realistiske opgaver med prototypen, fordi mængden af objekter var for lille. Det er derfor nødvendigt at benytte en forholdsvis stor datamængde for at kunne lave en ordentlig afprøvning eller overveje nøje, hvordan der kunne reduceres i mængden. En udvælgelse på baggrund af en speciel type emneord er alt for abstrakt til at brugerne kan abstrahere fra og overskue denne reduktion. Hvorimod det ville være lettere at forholde sig til et datagrundlag, der var reduceret til alle objekter fra 1985 og frem. I denne prototype vil jeg afgrænse datagrundlaget til alle danske objekter, hvilket er cirka 3 millioner. Denne afgrænsning er dels for at nedbringe behandlingstiderne i forbindelse med konstruktionen af prototypen, dels fordi de sproglige værktøjer, der inddrages, omhandler det danske sprog. Denne afgrænsning vil svare til, at der i de eksisterende søgesystemer, der benytter DanBib som datagrundlag,

³Den delmængde af DanBib jeg har til rådighed svarer til den datamængde der benyttes i den nye portal www.bibliotek.dk, hvilket er cirka 10 millioner referenceposter.

vælges at søge efter danske objekter. Det vil ikke være problematisk i forbindelse med afprøvning, fordi det er en afgrænsning, brugerne ofte anvender i forbindelse med deres søgning.

1.4 Problemformulering

Dette projekt skal belyse og analysere problemstillinger omkring forskning i fleksible søgesystemer. Ved at inddrage en kommunikativ vinkel skal der i forskningsprocessen tilføjes viden om målgruppen, der kan forbedre og understøtte de valg der træffes i designet af prototypen. En række datalogiske metoder skal undersøges og afprøves, på den valgte målgruppe, for at afgøre om de kan tilføre funktionalitet, der kan forbedre søgningen. Projektet skal overordnet svare på følgende spørgsmål:

Kan bibliotekarers behov, ønsker og krav til bibliografiske søgesystemer understøttes af et generisk intelligent søgesystem?

For at dette bliver muligt må det generisk intelligente søgesystem designes og implementeres, således at det kan afprøves af målgruppen, og derigennem give svar på spørgsmålet.

Udover at den kommunikative del skal bidrage til en større indsigt i målgruppen, bliver det interessante spørgsmål, hvilken indflydelse denne dimension får på udviklingsprocessen?

Kapitel 2

Metode

I dette kapitel beskrives de metodiske overvejelser. Først redegøres der for den kommunikative forforståelse, herefter beskrives den metode, der ønskes anvendt til udvikling af søgesystemet, og til sidst beskrives de datalogiske metoder og programmeringstekniske valg.

2.1 Kommunikativ forforståelse

Den traditionelle kommunikationsteori kan grundlæggende anskues ud fra to forskellige synsvinkler; kommunikation som enten en overførsel af et budskab eller som en udveksling og dannelse af meninger [Fiske, 1990]. Hver af synsvinklerne anvender metoder og antagelser, der stammer fra enten samfundsvidenskab eller humaniora, og man kan derfor også klassificere de to synsvinkler som værende to modstridende paradigmer, det samfundsvidenskabelige kontra det humanistiske [Sepstrup, 1999], eller skoler, processkolen kontra den semiotiske skole [Fiske, 1990].

Udgangspunktet for de to synsvinkler er hhv. afsenderens distribution og modtagerens reception. Grundlæggende bygger de begge på en simpel kommunikationsmodel, hvor en afsender sender et budskab gennem et medie til en modtager for at opnå et bestemt formål. Men hvor den samfundsvidenskabelige tilgang fokuserer på afsenderens måder og metoder til at opnå en tilstræbt/planlagt effekt hos modtageren, forholder det humanistiske paradigme sig til, hvordan modtageren opfatter budskabet, og ikke mindst til den mening, modtageren konstruerer ud fra den oplevede kommunikation og sine præferencer generelt.

Problemet med disse to modeller er, at de ikke på en naturlig måde kan bøjes, så de kan beskrive den mere dynamisk kommunikationsproces, der eksisterer i moderne medietyper, som udspringer fra computeren. De forhold-

der sig primært til den uopfordrede henvendelse, og ikke til kommunikation, der kræver aktiv deltagelse fra modtagerens side.

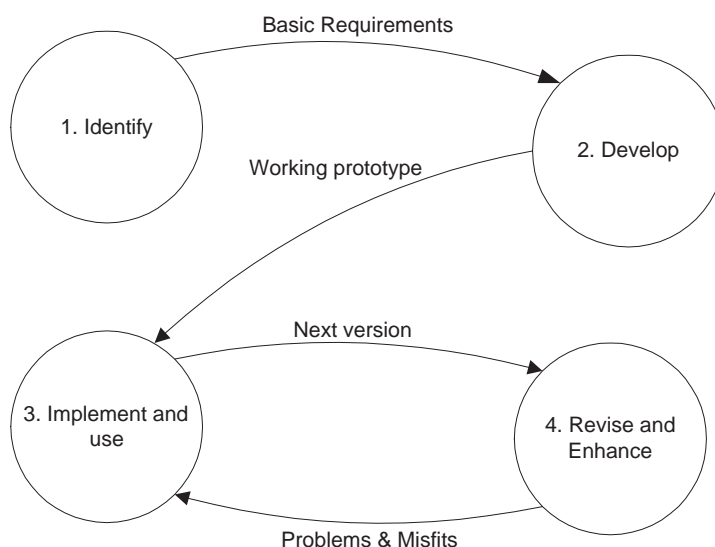
En anden teori "Uses and Gratifications" (U&G) fokuserer, selvom den udspringer indenfor det samfundsvidenskabelige paradigme, lige så meget som det humanistiske paradigme på modtagerens rolle i en vellykket kommunikationsproces. Men i stedet for at fokusere på, hvordan modtageren forstår budskabet, og hvorfor budskabet forstås på netop denne måde, fokuserer U&G på modtagernes brug af kommunikation som en nyttig måde til at opnå en behovstilfredsstillelse [Sepstrup, 1999]. Modtagerens adfærd er som regel målrettet, og man må, for at forstå informationsprocessen og måden, som modtageren anvender informationen på, som udgangspunkt forholde sig til det enkelte menneskes "*mål, behov, forudsætninger og begrænsninger*" [Sepstrup, 1999]. Et andet væsentligt element i U&G er, at den forudsætter aktive modtagere [Sepstrup, 1999] og derfor bedre kan beskrive den kommunikative forforståelse i dette projekt.

En af de primære dele i et søgesystem er netop den kommunikation, der foregår mellem brugeren og søgesystemet. Det kræver en vis form for engagement fra brugerens side. Denne proces beskrives ofte med begrebet interaktivitet. Den kan defineres som "*en proces, hvor to eller flere parter gensidigt påvirker hinanden gennem kommunikation*" [Andersen and Lindstrøm, 1997]. Det er altså modtagerens behov for information og evnen til interaktion med systemet, der bestemmer kvaliteten af kommunikationsprocessen mellem en bruger og et søgesystem. Indholdet i et søgesystem er bestemt af det datagrundlag, der søges i, og vil derfor ikke i sig selv være holdningsbærende. Det er heller ikke meningen, at der gennem indholdet skal skabes holdningsændringer hos målgruppen. Det bliver derfor udformningen, og grænsefladen, der bliver det bærende og formidlende i budskabet. Dette gælder både den visuelle og funktionelle del af grænsefladen. Succeskriteriet for kommunikationen i et søgesystem afhænger således af søgesystemets evne til at fremvise systemets muligheder og den fundne information, og i hvor høj grad de fundne objekter tilfredsstiller brugernes behov for information.

2.2 Systemudvikling

Den metode, jeg ønsker at anvende til design og udvikling af søgesystemet, kaldes *prototyping*. Prototyping er ikke en enkelt sammenhængende metode, men en samling teknikker der kan udnyttes ved systemudvikling [Frøkjær, 1985]. Formålet med denne metode er at konkretisere problemstillingerne i designet ved at benytte prototyper. Den første prototype skal helst etableres så hurtigt som muligt og danne basis for den videre udvikling. Det centrale i metoden er, at denne første prototype ikke blot er én blandt flere skitser, men

første prototype af det endelige system. Der skal derfor ikke introduceres et specifikt værktøj alene til kommunikation af designet, for eksempel proces- eller objektmodeller, fordi kommunikation af designet formidles gennem konstruktionen af systemet [Beyer and Holtzblatt, 1998]. Efter udvikling af den første prototype fortsætter processen som iteration mellem afprøvning hos brugerne, tilpasning og udbedring af fejl og videreudvikling af prototypen. I figur 2.1 ses en model der viser forløbet i prototyping [Naumann and Jenkins, 1982, side 29-44].



Figur 2.1: En skitse af processerne i prototyping-modellen.

Udviklingen af komplekse computersystemer er ofte meget abstrakt, og de forskellige delprocesser er afhængige af hinanden. Det er derfor vanskeligt at skille processerne fra hinanden for derigennem at simplificere kompleksiteten, uden at det får indflydelse på den delproces, man forsøger at beskrive. Med prototyping konkretiseres designet, og det bliver lettere for brugerne at give en brugbar kritik, fordi det er nemmere at fremkomme med alternative forslag til noget, man kan prøve. Hvis den problemstilling, man ønsker at brugerne skal forholde sig til, er meget abstrakt og kompleks, vil man ofte kun kunne få et ja/nej eller godt/dårligt svar, der ikke giver den fornødne indsigt til at afgøre, hvordan systemet skal ændres for at komme videre, eller hvorfor det eventuelt var godt [Beyer and Holtzblatt, 1998].

I en kommerciel udviklingsproces er målet at bruge kendte og afprøvede metoder og teknikker, til at løse problemer eller forbedre eksisterende løsninger. I dette projekt er målet forskning, og derfor at afprøve nye metoder og teknikker. Målgruppens rolle vil derfor være anderledes, og skal i højere grad bruges til vejledning og afprøvning end som mål for processen. Det betyder

ikke, at målgruppens behov og ønsker kan negligeres, hvis de er i modstrid med det, der skal undersøges, men heller ikke at centrale dele i forskningen skal undlades, hvis de ikke falder i målgruppens smag. I modsætning til den kommercielle udviklingsproces, hvor målgruppen er vanskelig at redefinere, fordi den ofte er en del af specifikationen for udviklingen, kan modsætninger mellem forskningens centrale dele og målgruppen i dette projekt skyldes, at det er en forkert målgruppe, der sigtes imod.

Udviklingen af den første prototype skal tage udgangspunkt i kendskab til målgruppen. Dette kendskab skal indledningsvis undersøges gennem en række interviews med en repræsentativ del af målgruppen. Analysen af disse interviews skal kondensere målgruppens ønsker og behov og bruges til at sikre, at det grundlæggende design i prototypen ikke udelukker mulighed for at opfylde målgruppens ønsker og krav, samt at den første prototype ikke er diametralt forskellig fra de søgesystemer, målgruppen normalt anvender.

2.3 Søgesystemet

I indledningen skitserede jeg to problemstillinger omkring søgesystemer, der ønskes undersøgt i dette projekt: behandling af objekter med forskellig struktur og generisk tilknytning af viden. Inddragelse af viden og intelligens, der bringer den i anvendelse i søgesystemer, gør systemerne mindre gennemskuelige, fordi det bliver vanskeligere at forstå, hvordan systemerne fremkommer med svaret. Hvis man, i det konventionelle søgesystem, har søgt efter objekter, der opfylder forespørgslen "A og B", ved man, at de objekter, der findes, opfylder netop denne forespørgsel. Et søgesystem kan blandt andet være fleksibelt ved at udvide en forespørgsel med elementer, der relaterer til de elementer forespørgslen indeholder. For eksempel kunne forespørgslen "A og B" blive udvidet til "(A, A₁, A₂) og (B)", fordi A og A₁ og A₂ er synonyme. Det kan derfor være svært at gennemskue, om det er A, A₁ eller A₂, der opfylder forespørgslen sammen med B. På den anden side vil systemet måske give bedre svar, netop fordi det selv formår at ekspandere forespørgslen. Prisen for at indføre en mere fleksibel forespørgselsevaluering i søgesystemer er uigennemskuelighed, så spørgsmålet bliver, om dette opvejes af de fordele, den fleksible forespørgselsevaluering giver?

Mange bibliografiske søgesystemer benytter boolesk logik med operatorerne *og*, *eller* og *ikke*¹. Boolesk logik er på samme tid simpelt, der er kun disse tre operatører, og komplekst, blandt andet fordi der mellem dem må indføres præcedensregler, der bestemmer, i hvilken rækkefølge de skal fortolkes. For

¹Dette er ikke kun gældende for bibliografiske søgesystemer, men for søgesystemer generelt.

eksempel kan forespørgslen ”*A og B eller C*” have to forskellige fortolkninger ”(*A og B*) eller *C*” eller ”*A og (B eller C)*”. Præcedensreglerne definerer normalt, at *og* binder stærkere end *eller*, hvilket betyder, at forespørgslen ”*A og B eller C*” implicit bliver til ”(*A og B*) eller *C*”. Ønskes den anden fortolkning, må der benyttes parenteser ”*A og (B eller C)*”.

Der findes forskellige teknikker, der kan evaluere forespørgsler, uden at der anvendes booleske operatører og derfor helt eliminere problemet med at skulle forholde sig til, hvordan de benyttes. Forskningsprojektet, *Fleksibel søgning i DanBib* [Andreasen and Bulskov, 1998], benytter en af disse teknikker, Ordered Weighted Averaging (OWA) [Yager, 1988, side 183-190]. Denne teknik har vist sig at være både velegnet og effektiv og er derfor oplagt at benytte i dette søgesystem.

2.3.1 Programmeringstekniske valg

Søgeselementet skal udvikles som en World Wide Web (WWW) applikation. Brugergrænsefladen skal udvikles i HyperText Markup Language (HTML), og datagrundlaget repræsenteres i en relationel database. Funktionaliteten skal programmeres i C/C++ og benytte Common gateway Interface (CGI).

Der benyttes i biblioteksverden en skelen mellem to forskellige typer af søgeapplikationer; tegnbaserede og web-baserede. De tegnbaserede applikationer er et levn fra tidligere tiders terminal-orienterede tankegang og benytter telnet-protokollen² til at kommunikere med søgesystemet. De eksisterer stadig i bedste velgående i biblioteksverden, trods det, at de har været nedprioriteret de sidste 5-6 år. Der er dog enighed om, at fremtidens søgesystemer skal være web-baserede, og det er også her ressourcerne bruges i dag. Web-baserede applikationer har deres styrke i at grænsefladen er velkendt og rimelig simpel at arbejde med. Der kan derfor hurtigt konstrueres prototyper, der indeholder mulighed for interaktion fra brugerne, hvilket gør web-baserede applikationer velegnede i forbindelse med prototyping. Ulempen ved web-baserede applikationer er de begrænsninger HTML-formatering har. Der udvikles mange forskellige teknikker til at overvinde nogle af begrænsningerne i HTML, men de skaber ofte nye problemer, primært i forhold til kompatibilitet og krav til brugernes browser [Nielsen, 2000]. Jeg valgte derfor at benytte ren HTML [Musciano and Kenney, 1998] med de begrænsninger, det indebærer.

Jeg har, gennem min tilknytning til ISL, arbejdet en del med relationelle databaser i forbindelse med fleksible søgesystemer, og valgte derfor at benytte den relationelle database til repræsentation af data. Dels ville jeg ikke selv

²Requests for Comments (RFC) 854.

skulle konstruere en database, og dels er det min erfaring, at den relationelle database fint kan opfylde behov for både overskuelighed og effektivitet. Roskilde Universitetscenter har en aftale med databasefirmaet Oracle, hvilket giver mig mulighed for at benytte Oracles database i dette projekt.

Der findes i dag mange forskellige erstatninger til CGI-programmering, for eksempel PHP, PERL, ASP et cetera, men fælles for dem er, at de er fortolkede programmeringssprog. I et system, der skal behandle store datamængder, er optimering essentielt, og fortolkede programmeringssprog er langsommere end oversatte programmeringssprog, alene af den grund, at de først på afviklingstidspunktet oversættes til binær kode. Det centrale i søgesystemet er kommunikationen mellem systemet og databasen, hvilket derfor vil være en flaskehals, der bestemmer, hvor effektivt søgningerne kan udføres. Jeg valgte at programmere søgesystemet i C/C++ [Stroustrup, 2000], [Josuttis, 1999], fordi grænsefladen til Oracle's relationelle database, Oracle Call Interface (OCI) [Belden and Melnick, 1999], fra dette sprog er utrolig effektiv og har gode muligheder for optimering.

Kapitel 3

Interviewundersøgelse

I dette kapitel beskrives den interviewundersøgelse der foretages for at afklare målgruppens ønsker og behov til søgesystemer. Grundlaget for udførelse af undersøgelsen samt overvejelser omkring validitet diskuteres. Der udarbejdes en interviewguide og de indsamlede data analyseres. Afslutningsvis konkluderes der på analysen. Transkription af interviewene findes på adressen www.isl.ruc.dk/giss eller www.styltsvig.dk/giss.

3.1 Interview

De indledende overvejelser i forbindelse med en interviewundersøgelse er tematisering og design, som skal lede til en formulering af en interviewguide [Kvale, 1994]. Denne fase skal klarlægge grundlaget for undersøgelsen, hvilket udbytte der forventes og hvordan dette opnås. Målet med den undersøgelse, der skal foretages i dette projekt, er at opnå indsigt i målgruppens behov, ønsker og krav til bibliografiske søgesystemer og deres holdninger til intelligente søgesystemer. Dette kræver, at der indhentes et vist kendskab til de værktøjer, der allerede anvendes, samt at der overvejes, hvilke mulige funktionaliteter der eventuelt kunne være interessante at inddrage under interviewet. Jeg har gennem mit arbejde i ISL tidligere stiftet bekendtskab med mange af de generelle værktøjer til bibliografisk søgning, der benyttes i det danske bibliotekssystem. Jeg har ligeledes en god indsigt i, hvilke udvidelser der kunne være interessante at inddrage i interviewet.

3.1.1 Repræsentativitet

Den viden, der indhentes gennem interviewet, skal benyttes som grundlag for de beslutninger, der skal træffes omkring design af prototypen. Dette gælder både for design af brugergrænsefladen og for det funktionelle design. Det er derfor nødvendigt at udvælge informanterne på en sådan måde, at den viden, der fremkommer fra analysen af de indsamlede data, kan gene-

raliseres. For at dette bliver muligt, skal informanterne være repræsentative for hele målgruppen. Repræsentativiteten kan sikres ved enten at udvælge informanterne helt tilfældigt eller, ud fra et kendskab til målgruppen, udvælge dem fra bestemte delmængder af målgruppen [Østbye *et al.*, 1997]. Det er svært på forhånd at gætte på, hvordan en repræsentativ delmængde udvælges, og hvor mange der skal til for at opnå repræsentativiteten. Det er ikke min vurdering, at en egentlig målgruppeanalyse er nødvendig, fordi gruppen er forholdsvist afgrænset. Samtidigt består opgaven i at understøtte vurderinger omkring tekniske og teoretiske valg, mere end at afklare noget endegyldigt omkring målgruppen. Et andet aspekt er, at udviklingsprocessen omkring prototypen skal foregå iterativt med målgruppen, forstået på den måde at den skal konsulteres flere gange gennem processen. Dette betyder, at eventuelle misforståelser eller fejlfortolkninger kan rettes løbende, ligesom der kan vælges andre delmængder af målgruppen på et senere tidspunkt. Omvendt vil valget af en ikke repræsentativ del af målgruppe betyde, at det senere kan blive nødvendigt at ændre på fundamentale dele af prototypen.

Den egenskab, der skal undersøges hos målgruppen, er deres behov, ønsker og krav til bibliografiske søgesystemer. Selvom søgeprocessen er en meget individuel proces, der kan betyde, at to bibliotekarer med samme arbejdsområde formentligt ikke angriber søgeprocessen på helt samme måde, vil der være en lang række dele af processen, der er ens eller ligner hinanden. Målet med interviewene er at klarlægge disse forskelligheder og ligheder og finde frem til et kompromis, der tilgodeser de fleste.

En første naturlig indsnævring er at afgrænse målgruppen til bibliotekarer, der er beskæftiget blandt Biblioteksstyrelsens interessenter¹, og som i deres daglige virke foretager bibliografiske søgninger. Dette sikrer, at den målgruppe, der udvælges blandt, har et praktisk forhold til bibliografiske søgninger. I denne afgrænsede målgruppe valgte jeg informanter fra tre forskellige grupper: folkebibliotekerne, forskningsbibliotekerne og resten, og to informanter fra hver af disse grupper. I gruppen "resten" valgte jeg at fokusere på DBC af flere grunde. Det er dem, der leverer datagrundlaget for projektet, og de har dermed en god indsigt og erfaring i søgning i netop denne datamængde. Jeg har gennem mit arbejde i ISL arbejdet sammen med DBC og ved, at de har en stor viden omkring bibliografiske søgesystemer. Deres indgang til søgning adskiller sig fra de to øvrige grupper, idet de ikke har den

¹Biblioteksstyrelsens væsentligste interessenter er det offentligt biblioteksvæsen, fordelt på en række nationale institutioner, en række større og mindre forsknings- og uddannelsesbiblioteker, samt de kommunale folkebiblioteker. De nærmeste samarbejdspartnere herudover er Kulturministeriet, andre ministerier, Kommunernes Landsforening, faglige organisationer, Dansk BiblioteksCenter samt institutioner og organisationer med tilgrænsende interesser.

daglige kontakt med lånerne. For de to andre grupper valgte jeg af praktiske grunde informanter fra henholdsvis Roskilde Bibliotek og Roskilde Universitetsbibliotek.

Jeg valgte ikke at stille andre krav til udvælgelsen af informanter i de enkelte institutioner, end at de tilhørte målgruppen, altså at de i deres daglige virke foretager bibliografiske søgninger. Det er derfor, for min del, helt tilfældigt, hvilke bibliotekarer på de enkelte institutioner, jeg kom til at interviewe. Hvilke kriterier de har brugt, kan jeg derfor kun gætte på, men under interviewene fik jeg alligevel en ide om, at der i alle tre grupperinger havde været overvejelser omkring at sikre en vis forskellighed mellem de to udvalgte informanter med hensyn til alder, erfaring og arbejdsområder.

3.1.2 Gyldighed og pålidelighed

Videnskabelige undersøgelser anvendelighed afhænger af deres gyldighed og pålidelighed². Interviewene blev derfor overordnet opbygget efter et skema, der indledningsvis skulle skabe tryghed i forhold til interviewsituationen, dernæst skabe indblik i informantens nuværende brug af bibliografiske søgeværktøjer og afslutningsvis prøve at introducere nye elementer. På denne måde skulle det sikres, at informanterne i begyndelsen af interviewet kunne føle sig afslappet og dermed ikke være fokuseret på situationen. Dette skulle medvirke til, at det var informanternes egne holdninger, der kom til udtryk, og ikke et forsøg på at tækkes interviewerens.

Det var min forventning, at der skulle bruges ca. en time pr. interview. Udformningen af spørgsmålene skulle være meget åbne, da jeg ellers ikke ville kunne sikre mig den fornødne indsigt, idet det er umuligt at få et nuanceret billede af behov, ønsker og krav, hvis informanten skulle begrænses af lukkede spørgsmål. Interviewene skulle optages på bånd, så jeg ikke var bundet af at tage notater, og derfor kunne koncentrere mig om at få interviewet til at udvikle sig, som jeg ønskede det. Dette øger også reliabiliteten, fordi en optagelse er væsentlig mere nuanceret end noter taget under interviewet.

Interviewene blev afholdt på informanternes "hjemmebane", idet det var mig, der tog afsted for at interviewe dem. De havde derfor selv bestemt, hvor og under hvilke omstændigheder interviewet skulle foregå. De havde på forhånd fået af vide, at interviewet ville vare omkring en time, og at indholdet skulle handle om søgning i bibliografiske databaser. Forud for

²Gyldighed eller validitet har at gøre med, hvorvidt man forholder sig til det, man ønsker at forholde sig til. Det vil sige, i hvilken grad design og operationaliseringer giver relevante indsigter i forholdet til den overordnede problemstilling. Eksempelvis svækker det validiteten, hvis informanterne under interviewet ikke opfører sig, som de ville have gjort uden for interviewsituationen. [Østbye *et al.*, 1997, Kapitel 5]

interviewet blev informanterne gjort opmærksomme på, at de i projektet kun ville optræde anonymt, at båndoptageren skulle benyttes for at give mig mulighed for at analysere interviewet efterfølgende, og at båndet ville blive slettet når projektet var afsluttet. Herefter fik informanterne en kort introduktion til projektet.

3.1.3 Interviewguide

Spørgsmålene i interviewundersøgelsen skal, som tidligere nævnt, være meget åbne, derfor blev der ikke udarbejdet en stringent interviewguide med alle de spørgsmål, der skulle stilles. Interviewguiden blev i stedet opdelt i tre faser med tilhørende spørgsmål, der primært skulle benyttes som rettesnor for kronologien og som huskeliste for, hvilke punkter jeg skulle berøre under interviewene.

Den indledende fase

I denne fase var målet at skabe ro hos informanten og indsamle viden omkring deres arbejde med bibliografiske søgninger. Denne viden skulle bruges i de to næste faser til at afgøre, hvordan de enkelte elementer blev belyst. Ved at få en indsigt i informantens arbejdsområde kunne jeg undgå at komme til at stille irrelevante eller meningsløse spørgsmål.

De overordnede spørgsmål til denne fase var følgende:

- Hvor lang tid har du arbejdet som bibliotekar?
- Hvad er dit primære arbejdsområde og hvad indebærer det i forhold til bibliografisk søgning?
- Hvilke medier benytter du i forbindelse med søgning? Inddrager du for eksempel bibliografier, ordbøger eller andet i forbindelse med definition af forespørgsler?
- Er der en sammenhæng mellem de opgaver, der skal løses og valget af databaser?

Det nuværende brug

Denne fase er utrolig væsentlig for udviklingen af prototypen, fordi det er her jeg kan opnå forståelse og indsigt i, hvordan informanterne benytter de eksisterende søgesystemer.

De overordnede spørgsmål til denne fase var følgende:

- Benytter du webben? Hvilke(n) søgeportale(r) benytter du og hvorfor? Hvilke parametre styrer dit valg?

- Hvilke elementer i de bibliografiske objekter vælger du oftest at søge efter (emneord, titler, klassifikation, et cetera)?
- Starter du i en bestemt database hver gang, eller er det afhængigt af konteksten?
- Hvis en forespørgsel er for snæver eller for bred, hvordan kommer du så videre derfra?
- Benytter du CCL³ og hvordan?
- Hvor mange termer benytter du i gennemsnit i forespørgsler og hvorfor?
- Benytter du oftest tegn-baserede eller web-baserede applikationer?
- Hvilke funktionaliteter benytter du?
- Hvordan skulle en web-baseret grænseflade designes, hvis du skulle bestemme (forespørgselsdefinition og svar)?

Det fremtidige søgesystem

I denne fase skal informanterne forholde sig til forskellige muligheder for at gøre søgesystemet mere intelligent.

De overordnede spørgsmål til denne fase var følgende:

- Hvordan er den overordnede holdning til intelligente søgesystemer?
- Hvad er holdningen til ekspansion af forespørgsler (det at systemet udvider forespørgslen med termer det finder relevante)?
- Hvis sådanne avancerede teknikker skal benyttes, hvordan skal grænsefladen så være?

3.2 Interviewanalyse

Analysen af de indsamlede informationer skal lede til en forståelse af to overordnet problemstillinger; hvordan benyttes de nuværende værktøjer og hvordan forholder informanterne sig til intelligente søgesystemer. Udfra hver af disse to problemstillinger kan der kondenseres elementer, som dels svarer til spørgsmålene i interviewguiden, og som dels er fænomener, der er opstået under interviewet. Denne analysemetode kaldes meningskondensering, og benyttes til at sammentrække udtrykte meninger til korte formuleringer [Kvale,

³Common Command Language, se afsnit 4.3.1 på side 38.

1994, Kapitel 11]. Det væsentligste i valget af denne analysemetode er, at det kan reducere de indsamlede data til en række korte formuleringer, der kan benyttes i designet af søgesystemet.

Informanterne var opdelt i tre grupper: Roskilde Bibliotek (RB), Roskilde Universitetsbibliotek (RUb) og Dansk BiblioteksCenter (DBC). Denne gruppering viste sig tydeligst i forhold til informanternes arbejdsområder. Informanterne fra RB havde publikumskontakt som det primære arbejdsområde, det at hjælpe lånere med løsning af forskellige problemer. Informanterne fra RUb havde publikumskontakt i en trediedel af deres arbejde. Mens informanterne fra DBC ingen egentlig publikumskontakt havde, men beskæftigede sig med undervisning i bibliografisk søgning. De var dog alle dagligt beskæftiget med bibliografisk søgning, og derfor alle en del af den målgruppe, der var defineret for undersøgelsen.

3.2.1 Den nuværende brug

Her sammenfattes de indhentede informationer om informanternes nuværende brug i forbindelsen med bibliografiske søgninger. De valgte kategorier er dels de spørgsmål, der er defineret i interviewguiden, og dels fænomener, der er udledt af selv analysen.

Brug af webben

Alle informanter anvendte søgesystemet Google⁴. Argumenterne for at benytte Google var alle informanter enige om. Den giver gode svar, ved at rangere de bedste svar først, grænsefladen er overskuelig og ikke fyldt med reklamer og den er hurtig.

Forespørgselsdefinition

Der viste sig to forskellige indgange til søgningen. En såkaldt *quick and dirty*-søgning, hvor man starter med en fritekstssøgning på et par termer for at afsøge området. Udfra dette svar arbejdes der videre. Den anden mulighed var at udtrykke komplekse forespørgsler i CCL. Alle informanter benyttede *quick and dirty*-søgning til at starte med. Herfra var der lidt forskel på, hvilken strategi de benyttede. De fleste anvendte søgessæt⁵ til at bevæge sig rundt i emnet. På den måde var det nemmere at gå ud af en sti med simple forespørgsler og fortryde og vende tilbage til et af søgesættene fra tidligere og vælge en anden vej. Nogle enkelte ville i bestemte situationer, hvis de var ret sikre på, hvordan de skulle komme videre, anvende komplekse forespørgsler.

⁴www.google.com

⁵Se afsnit 4.3.1 på side 38.

Alle informanter anvendte CCL i deres søgning. Nogle mere end andre. Men ingen anvendte de mere specielle aspekter i CCL-sproget. Den primære anvendelse var booleske operatører og muligheden for at benytte et præfiks til at bestemme, i hvilken del af objekterne der skulle søges. For eksempel $F0=peter$, der definerer, at termen *peter* skal være en forfatter.

Når informanterne skulle vælge, hvilke dele af objekterne de oftest søgte på, var det enten emneord, forfatter eller klassifikation. Alle informanterne anvendte mellem 1 og 3 termer i forespørgslerne. I sjældne tilfælde flere termer, men i så fald som en *eller*-kombination. For eksempel ville det ved en søgning efter emnet *email* være oplagt at benytte alle de forskellige staveformer *email*, *e-mail*, *e-post*, et cetera.

Generelt kan man konkludere at de søgestrategier, de anvendte, var så tæt på hinanden, at det ikke kunne være en tilfældighed. Det er naturligvis et udtryk for den skoling der foretages under bibliotekaruddannelsen, og man kan konkludere, at de alle følger den meget konsekvent. Hvilket kan være udtryk for at metoden er god og effektiv.

Applikationer

Alle informanter foretrak tegnaserede applikationer frem for web-baserede. Dette skyldtes flere ting. Først og fremmest er de tegn-baserede applikationer hurtigere, og hvis man kan anvende CCL, har de en større udtrykskraft. Et andet argument var, at de fleste web-baserede applikationer er udviklet til brug for slutbrugerne, altså lånerne, og derfor ikke har samme muligheder som de tegnaserede. De er for eksempel ikke integrerede med administrationssystemerne, det er svære at arbejde med søgesæt og visningsformaterne er ikke lige så nuancerede.

Søgefunktionalitet

Der var ingen af informanterne, der anvendte specielle søgefunktionaliteter, udover *scan*⁶. De fleste vidste ikke, hvad *zoom*⁷ og *relevans feedback*⁸ var, og når det blev forklaret, mente de, at det kunne være spændende, men det var ikke noget, de benyttede. Der var en tendens til, at det var de ældre

⁶Scan er en funktion, hvor man kan skanne en liste efter specifikke elementer. Er man for eksempel ikke sikker på, hvordan en forfatters navn staves, kan en liste med alle forfattere skannes for at afgøre, hvordan navnet staves, eller hvis man kun kan huske starten af en titel, kan en titelliste skannes, og den ønskede titel udvælges [DBC, 2001].

⁷Zoom er en funktionalitet, der på baggrund af de fundne objekter i et svar, benytter alle, for eksempel emneord, i svaret til en ny forespørgsel.

⁸Relevans feedback er en funktionalitet, hvor brugeren ved at vælge en eller flere objekter i svaret beskriver, hvilke objekter der er relevante. Søgssystemet benytter herefter de valgte objekter til at danne en ny forespørgsel ud fra indholdet [Salton, 1988].

bibliotekarer, der anvendte *scan*.

Ønsker til en web-grænseflade

Alle informanter var enige i, at de tegn-baserede applikationer inden for de nærmeste år forsvinder helt. De har allerede de sidste 4-5 år været nedprioriteret og på vej ud, og derfor ikke udviklet sig. Der var derfor ikke uenighed om, at man i forhold til fremtidige systemer skulle fokusere på web-baserede applikationer. Overordnet var informanternes ønske til en web-grænseflade, at den er hurtig, simpel og kan det hele. De var dog alle klar over det modsætningsforhold, der er mellem den enkle grænseflade og komplekse funktioner.

Grænsefladen skal kun have et indtastningsfelt, hvor der kan benyttes CCL. Det skal være muligt at vælge forskellige visningsformater og antal viste objekter. Udprintningsfunktionen skal give mulighed for, at der kan vælges forskellige objekter, som senere udskrives. Mulighed for at opsamle informationer undervejs, for eksempel ord, klassifikation, et cetera, ville være kærkommen, idet dette nu foretages på papir.

3.2.2 Det fremtidige søgesystem

Informanterne var i forbindelse med holdningen til intelligente systemer delt. Informanterne fra DBC havde ikke samme modstand mod avancerede søgesystemer som resten af informanterne. Men generelt var der en lidt konservativ holdning til søgesystemer blandt informanterne. De var dog alle enige om, hvad der for dem var problematisk ved avancerede søgesystemer, nemlig manglen på kontrol og gennemsøkelighed. Som udgangspunkt er der derfor ikke grundlag for at tilbyde bibliotekarer avancerede søgesystemer.

Generelt var der også en negativ holdning til ekspansion af forespørgsler. Hvis søgesystemet ændrer på forespørgslen ved at tilføje termer, mistes overskueligheden, og det bliver vanskeligt at gennemsøge, hvordan et svar er fremkommet, og hvordan man kommer videre, hvis svaret ikke er godt nok. Der var derimod en generel positiv holdning til at benytte den viden, ekspansionen benytter til opslag, i forbindelse med definering af forespørgsler. Altså som viden man kan trække på i forbindelse med definition af forespørgsler. En funktionalitet, hvor man kan få beskrevet, hvilke termer der knytter sig til en bestemt term, ville være en funktion, alle informanterne kunne forstille sig at have behov for.

3.2.3 Opsamling

Umiddelbart kunne konklusionen være, at denne målgruppe hverken har behov for eller ønsker avancerede søgesystemer. Der dog nogle modsætninger mellem deres holdning til avancerede søgesystemer og deres søgestrategi. De

starter alle, uden undtagelse, med en såkaldt *quick and dirty*-søgning, der handler om at belyse et bestemt emne. Denne belysning, eller undersøgelse, ville, efter min mening, understøttes bedre ved at benytte avancerede søgesystemer, der ikke er begrænset af boolesk logik. Hvor det er muligt at beskrive emnet med alle de informationer, man har om det, og lade systemet om at udvælge de objekter, der bedst opfylder dette.

Et andet modsætningsforhold ved informanternes modvilje mod avancerede søgesystemer er deres brug af Internet-søgeportaler. De anvender alle Google, fordi den er bedre til at finde relevante objekter og viser de mest relevante objekter først. Dette er ikke muligt uden avancerede søgeteknikker.

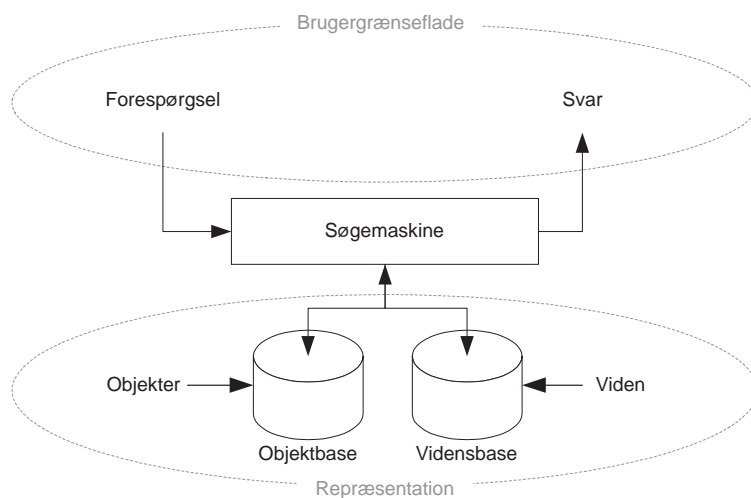
Den del af informanterne, der havde mest publikumskontakt, mente, at en funktionalitet, der på baggrund af lånerens viden, kan finde objekter der ligner denne beskrivelse, ville være en stor hjælp. Dette er heller ikke muligt uden avancerede teknikker.

Jeg har fuld forståelse for informanternes bekymring omkring overskuelighed, og overser ikke behovet for verifikation, men den lidt firkantede holdning til avancerede søgesystemer, hænger ikke sammen med nogle af de behov, de udtrykte. Jeg vil derfor fokusere på denne del i den første prototype og efterprøve om avancerede teknikker kan afhjælpe nogle af de problemstillinger, de beskriver. Specielt er det interessant at undersøge, hvordan målgruppen vil modtage avancerede teknikker i den første søgefase; *quick and dirty*-søgningen.

Kapitel 4

Design af prototype

I dette kapitel beskrives designet af prototypen¹. Indledningsvis beskrives strukturen i datagrundlaget for at give det fornødne grundlag for at forstå de valg, der baserer sig herpå. Herefter konstrueres en datamodel til repræsentation af objekter, indeksering og viden. Brugergrænsefladen og forespørgselsprog behandles, og afslutningsvis designes et associationsnet og forespørgselsevalueringen.



Figur 4.1:

I figur 4.1 ses en overordnet skitsering af søgesystemet. Systemet har tre forskellige input: forespørgsler, objekter og viden. Den øverste del af figuren beskriver grænsefladen mellem brugerne og søgemaskinen, og den nederste del grænsefladen mellem repræsentation af objekter og viden og

¹Der vil gennem afsnittet blive refereret til prototypen og søgesystemet som en og samme ting.

søgemaskinen. Søgemaskinen er kernen i søgesystemet. Det er her systemet bindes sammen, og hvor grænsefladerne til brugerne og repræsentationen defineres. Søgesystemet kan derfor opdeles i tre overordnede moduler; brugergrænseflade, søgemaskine og repræsentation. Ved at definere grænseflader mellem disse moduler gøres de uafhængige af hinanden, og ændringer i et modul behøver derfor ikke involvere de andre. Denne modularitet skal gøre systemet mere fleksibelt og lettere at udvikle.

4.1 Datagrundlag

Datagrundlaget i prototypen er, som tidligere nævnt, det udsnit af DBC's DanBib, som benyttes i internetportalen *bibliotek.dk*. Objekterne i datagrundlaget er referencer til bibliografiske enheder², det vil sige de beskriver de enheder, de refererer til. Objekter i DanBib er formateret efter et format der hedder danMARC2 [Katalogdatarådet for Biblioteksstyrelsen, 1998a]. DanMARC2-formatet er en standard for bibliografiske dataposter i maskinlæsbar form, som er baseret på de katalogiseringsregler, der anvendes på de danske biblioteker [Katalogdatarådet for Biblioteksstyrelsen, 1998b]. Formålet med denne formateringsstandard er at kunne udveksle bibliografiske informationer.

```
001 00*a0747561*b820050*fa
002 00*d8778382424
004 00*rn*ae
008 00*tm*a1997*1dan*v4
009 00*aa*gxx
021 00*a8778382424
088 00*a027.1*bPrivatbiblioteker*1(489) Danmark
089 00*a02(489)(091)*bBibliotekshistorie, Danmark
096 00*aL2 BIB*sStue*tBøger*b01*iKj*o027.1
100 00*aKjølsten*hKlaus
245 00*aHendes Majestæt Dronningens Håndbibliotek : 1746-1996*pHer
    Majesty the Queen's reference library*eAbridged version by
    Christian Gottlieb
260 00*aOdense*bOdense Universitetsforlag*c1997
300 00*a336 s.
631 00*aPrivatbiblioteker - Danmark - Dronningens Håndbibliotek
631 00*aBibliotekshistorie - Danmark - Dronningens Håndbibliotek
x45 00*aHendes Majestæt Dronningens Håndbibliotek : 1746-1996
m01 00*ate
m06 00*abä
```

Figur 4.2: Et eksempel på en danMARC2-formateret post.

Et objekt i danMARC2-formatet er opdelt i felter og delfelter. Felterne defineres af et 3-cifret feltnummer³. Hvert felt indeholder et eller flere delfelter,

²Bibliografiske enheder kan være bøger, CD'er, billeder(kunst), artikler, tidsskrifter et cetera.

³Plus en efterfølgende indikator, som benyttes ved inddatering, og derfor vil være 00 i de eksempler, der vises her.

der defineres med 2 tegn, hvor det første er *, og hvor det andet normalt er et lille bogstav, for eksempel *a, *b eller *h.

I figur 4.2 vises et eksempel på et danMARC2-formateret objekt. Den første linie er feltet 001, som indeholder delfelterne *a0747561, *b820050 og *fa. DanMARC2-formatet er et linieformat med et felt på hver linie. Overstiger linielængden 80 tegn, opsplittes den i flere liner. I figur 4.2 er felt 245 opsplittet i 3 liner. Linierne, der er splittet, begynder med 4 blanke tegn og kan sammensættes under læsningen af data til en linie.

Delfelt	Gentages	Beskrivelse
t		kode for bibliografisk kategori
u		kode for udgivelsesstatus
a		udgivelsesår
z		efterfølgende udgivelsesår
b	G	kode for udgivelsesland
c		bogstavkode for et periodicums frekvens
d	G	kode for indholdets form
e		kode for offentlig publikation
f		kode for konferencepublikation
g		kode for festskrift
h		kode for periodicumtype
i		kode for hovedtitlens alfabet eller skriftsystem (issn)
j		kode for skønlitterær form
k		kode for biografi
l		kode for hovedsprog
m		kode for stor skrift
o		kode for børne- eller skolemateriale
q	G	kode for filtype
r		kode for værtspublikationens type
v		kode for katalogiseringsniveau

Tabel 4.1: Delfelterne, der kan tilknyttes feltet 008. Kolonnen *Gentages* definerer, hvorvidt delfeltet kan optræde mere end en gang i et felt.

Felterne fra 000 til 999 er fastlagte og defineret i [Katalogdatarådet for Biblioteksstyrelsen, 1998a] med tilhørende delfelter. Et felt kan have mange forskellige delfelter, hvilket betyder at detaljeringsgraden er meget stor. I tabel 4.1 vises de delfelter, der kan tilknyttes felt 008, *Generelle søgekoder for bibliografiske materialer*, for at illustrere detaljeringsgraden. Der kan lokalt tilknyttes felter, blot de ligger uden for intervallet 000 til 999. I figur 4.2 er felterne x45, m01 og m02 lokale felter tildelt af DBC.

4.2 Repræsentation

Et af målene for det søgesystem, der skal udvikles i dette projekt, er at skabe en intern repræsentation, der dynamisk kan behandle objekter med forskellig struktur. DanMARC2-formatet er naturligvis en oplagt kandidat, fordi den høje detaljeringsgrad sikrer, at mange forskellige objekter med forskellige strukturelle formater kan beskrives indenfor formatets ram-

mer. Et problem med danMARC2-formatet er, at det er et referenceformat, der repræsenterer beskrivelser af objekter og ikke objekterne selv. Selvom danMARC2-formatet med de nuværende rammer kan repræsentere mange forskellige objekttyper, er det en begrænsning, at det kun er referencer til objekterne, der repræsenteres. Der er behov for også at kunne repræsentere selve objektet, for eksempel hele artikler. Teknisk set er det muligt at definere lokale felter, som kan repræsentere brødtekst uden at komme direkte i konflikt med danMARC2-formatet. Det vil dog være et brud på intentionen med danMARC2-formatet at udvide det, med mulighed for at repræsentere objekter, der ikke er en reference til objektet, men objektet selv. Jeg valgte at konsultere DBC for at finde ud af, om de mente, der ville være problemer forbundet med denne udvidelse. Det var der ikke, så jeg valgte at benytte danMARC2-formatet som grundlag for repræsentation af data i søgesystemet.

Offentliggjort 06.06 2001 00.16

Chilensk dommer vil afhøre Kissinger

En chilensk undersøgelsesdommer vil sende en begæring til den tidligere amerikanske udenrigsminister Henry Kissinger om at afhøre ham som følge af mordet på den amerikanske statsborger Charles Horman, der skete umiddelbart efter det blodige militærkup i Chile i 1973. Det oplyser juridiske kilder i Chile tirsdag ifølge det spanske nyhedsbureau EFE.

Dommer Juan Guzmán, som undersøger en række sager mod den tidligere chilenske diktator Augusto Pinochet, har en række spørgsmål til Kissinger om begivenhederne, der kulminerede med drabet på Horman.

Sagen var inspiration for Costa-Gavras-filmen "Savnet" med Jack Lemmon i rollen som Charles Hormans far.

Kissinger betragtes som impliceret, fordi Charles Horman undersøgte den amerikanske efterretningstjeneste CIA's rolle i forbindelse med kuppet, da han blev tilbageholdt af de militære myndigheder. . .

Figur 4.3: Et udsnit fra en artikel (Fra Internetavisen Jyllandsposten).

Fuldtekstobjekter vil, i modsætning til referencer til objekter, ikke have en stringent strukturel opbygning, men vil være opbygget på baggrund af, hvad teksten skal formidle. Artikler vil dog ind mellem benytte en struktur, hvor første afsnit er en slags abstrakt der beskriver indholdet i artiklen. Jeg valgte derfor, at der i repræsentationen skulle differentieres mellem to typer af brødtekst, normal og indholdsbeskrivende. Med denne differentiering kan brødteksten behandles forskelligt i søgesystemet. For eksempel kan termer, der optræder i den indholdsbeskrivende del, vægtes højere. Skulle det vise sig senere, at der er behov for yderligere differentiering, vil dette ikke kræve, at repræsentationen skal redefineres, så der er ikke langtrækkende konsekvenser forbundet med dette valg. DanMARC2-formatet udvides derfor med feltet `btx`, med delfelterne `*a` til indholdsbeskrivende brødtekst og

***b** til normal brødtekst.

I figur 4.3 vises et udsnit af en artikel fra Internetavisen Jyllandsposten. Det første afsnit er med fed skrift for at illustrere, at det er indholdsbeskrivende brødtekst. I figur 4.4 er artiklen konverteret til det udvidet danMARC2-format. Feltet 008 00*a2001*bdk*ldan definerer at udgivelsesåret er 2001, at udgivelseslandet er danmark og at sproget er dansk. Felt 245 er titlen. I DanBib benytter DBC felterne m01 - m06 til lokalt definerede materialebeskrivelser. Jeg definerer feltet mat til materialebeskrivelse for fuldtekstobjekter. I eksemplet ai=avisartikel. Feltet btx har 2 indgange. En for hvert delfelt⁴.

```
008 00*a2001*bdk*ldan
245 00*aChilensk dommer vil afhøre Kissinger
btx 00*aEn chilensk undersøgelsesdommer vil sende en begæring ti=
    l den tidligere amerikanske udenrigsminister Henry Kissinger
    om at afhøre ham som følge af mordet på den amerikanske stat=
    sborger Charles Horman, der skete umiddelbart efter det blod=
    ige militærkup i Chile i 1973. Det oplyser juridiske kilder =
    i Chile tirsdag ifølge det spanske nyhedsbureau EFE.
btx 00*bDommer Juan Guzmán, som undersøger en række sager mod de=
    n tidligere chilenske diktator Augusto Pinochet, har en rækk=
    e spørgsmål til Kissinger om begivenhederne, der kulminerede
    med drabet på Horman.\n
    Sagen var inspiration for Costa-Gavras-filmen "Savnet" med J=
    ack Lemmon i rollen som Charles Hormans far.\n
    Kissinger betragtes som impliceret, fordi Charles Horman und=
    ersøgte den amerikanske efterretningstjeneste CIA's rolle i =
    forbindelse med kuppet, da han blev tilbageholdt af de milit=
    ære myndigheder...
mat 00*aai
```

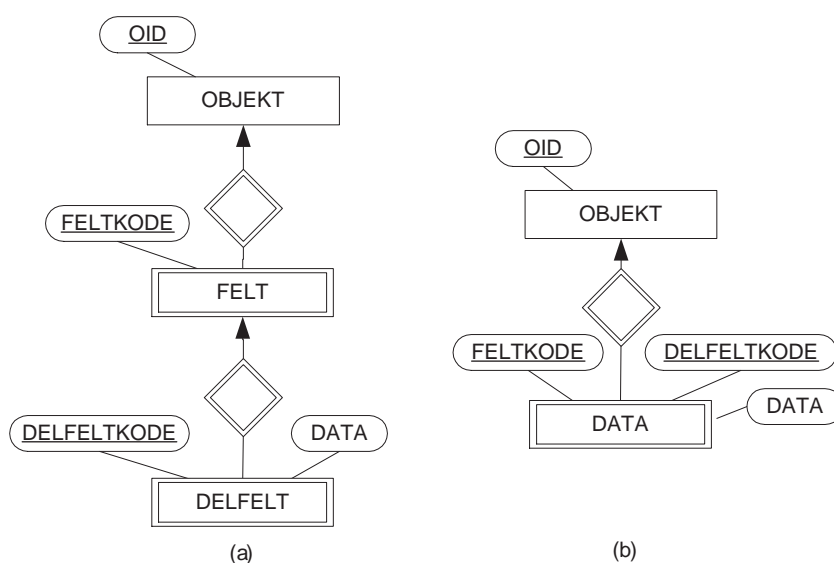
Figur 4.4: Et eksempel på en danMARC2-formateret post.

Denne udvidelse af danMARC2-formatet muliggør repræsentation af objekter med meget forskellig struktur. Hvis søgesystemet understøtter det udvidede danMARC2-format, vil der dynamisk kunne tilknyttes mange forskellige objekttyper. Det eneste krav til objekterne er, at de kan konverteres til dette format.

4.2.1 Objekter

Repræsentationen af danMARC2-formatet er forholdsvis simpel. Hvert objekt har et eller flere felter, der hver især kan indeholde et eller flere delfelter, hvilket er en hierarkisk struktur i tre niveauer. I figur 4.5a vises en datamodel til repræsentation af danMARC2-formatet, hvor de tre niveauer i hierarkiet modelleres med hver sin entitet. Der er imidlertid ikke behov for, at FELT

⁴Det er ikke nødvendigt at opsplitte feltet btx med en linie til hvert delfelt. Det er gjort her for at skabe overskuelighed.



Figur 4.5: To eksempler på datamodeller, der kan benyttes til at repræsentere danMARC2-formatet.

har en selvstændig entitet, og derfor kan modellen forenkles, som det er vist i figur 4.5b.

4.2.2 Metainformation

I forbindelse med søgning vil objekternes struktur kunne bruges i brugergrænsefladen eller forespørgselsprog til at bestemme, hvilke dele af objekterne der skal afsøges. Det vil dog være for uoverskueligt, hvis danMARC2-formatet ikke forenkles, fordi det er umuligt for brugeren at huske tusindvis af forskellige felt/delfelt kombinationer. Derfor grupperes felter/delfelter sammen i grupper, der kaldes *søgekoder*, og som kan benyttes i forbindelse med søgningen til at afsøge en gruppe felter/delfelter. Biblioteksstyrelsen har defineret ”Praksisregler for søgeveje” [Biblioteksstyrelsen, 1999], der definerer sammenhængen mellem en række søgekoder og danMARC2-formatet.

I tabel 4.2 vises, hvilke felter/delfelter der definerer søgekoden *cl*, *Klassifikation*. Formålet med at opsætte praksisregler for søgekoder er at sikre den størst mulige ensartethed i anvendelse af søgekoder i forhold til danMARC2-formatet, således at samme søgekode, anvendt i forskellige søgesystemer, vil afsøge samme dele af objekterne og

Felt	Delfelt
087	a
088	a
089	a
089	a
652	i, m, n, o, p, q, r
654	i, m, n, o, p, q, r

Tabel 4.2: Definitionen af søgekoden *cl*, *Klassifikation*, i felter og delfelter.

dermed fremkomme med samme svar på samme datagrundlag.

- **cl** - Klassifikation, alle koder
 - **dk** - DK5
 - **ok** - Klassifikation som opstilling
 - **gd** - Forældet dk-klassemærke
 - **kl** - Lokal klassifikation

Figur 4.6: Hierarkisk opbygning af søgekoder for klassifikationsinformationer.

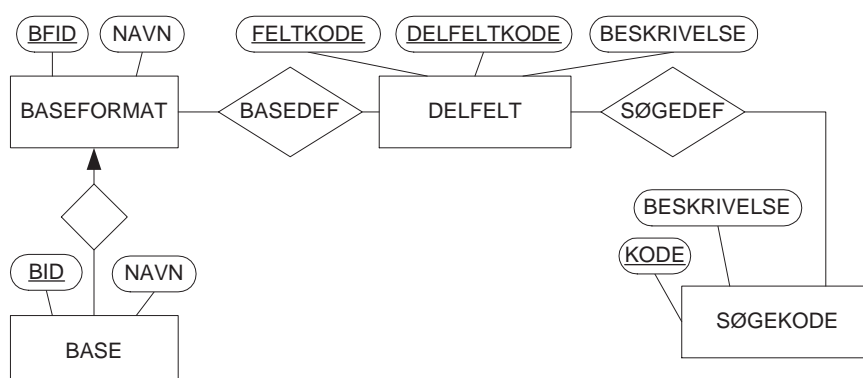
Søgekodedefinitionen er opbygget hierarkisk. I figur 4.6 vises et eksempel på den hierarkiske opbygning af søgekoder for klassifikation. Informanterne i interviewundersøgelsen var alle enige om, at de benyttede søgekoder i deres forespørgsler, men de søgekoder, de nævnte, tilhørte alle det øverste niveau i hierarkiet⁵. Der er for enkeltordssøgekoder⁶ defineret 65 forskellige koder for hele hierarkiet, hvoraf 23 er på øverste niveau.

Ved at benytte søgekoderne til gruppering af informationer i danMARC2-formatet opnås der dels en understøttelse af brugernes ønsker og behov og dels en større fleksibilitet. Behandlingen af objekterne i søgesystemet defineres i forhold til søgekoderne, således at alle data, der er grupperet under en søgekode, behandles på samme måde. Hvis der senere skal tilføjes flere informationer, kan de enten tilføjes under en eksisterende søgekode, eller der kan defineres en ny søgekode med tilhørende behandlingsstrategi. Systemet skal ikke behandle information på felt/delfelt-niveau, og vil derfor være fleksibelt.

Datamodellen i figur 4.7 viser hvordan metainformationerne i danMARC2-formatet skal repræsenteres i søgesystemet. Entiteten SØGEKODE repræsenterer de mulige søgekoder i systemet. Hver søgekode definerer en mængde felter/delfelter gennem relationen til entiteten DELFELT, der definerer de mulige felter/delfelter. For at generalisere metadefinitionen valgte jeg at indføre muligheden for at repræsentere flere forskellige formater, der har det tilfælles, at de kan defineres ved felter/delfelter. Entiteten BASEFORMAT indeholder de forskellige formater, der repræsenteres i søgesystemet. I denne prototype vil det kun være DanMARC2-formatet, der repræsenteres. Entiteten BASE definerer en konkret datamængde, her i prototypen en delmængde af DanBib.

⁵Dette kan skyldes, at jeg ikke bad dem forholde sig eksplicit til denne problemstilling, men blot kom ind på det i forbindelse med forespørgselsdefinition i almindelighed.

⁶Søgekoderne er opdelt i to overordnede grupper; enkeltordskoder og langordskoder. Enkeltordskoder er defineret for delfelter i felterne, mens langordskoder ofte definerer en kombination af delfelter, der til sammen danner en ny betydning.



Figur 4.7: Datamodel af repræsentation af metainformationer.

I tabel 4.3 vises de søgekoder, jeg udvalgte til brug for dette søgesystem. De er en udvalgt delmængde af de søgekoder, Biblioteksstyrelsen har defineret, plus koder for den udvidelse, jeg har tilføjet, samt en eksplicit defineret af et vokabularium *vo*. Dette vokabularium benyttes til at gruppere og udvælge den delmængde af felter og delfelter i feltintervallet 600 – 699, *Emnedata*, der kan benyttes til etablering af et associationsnet.

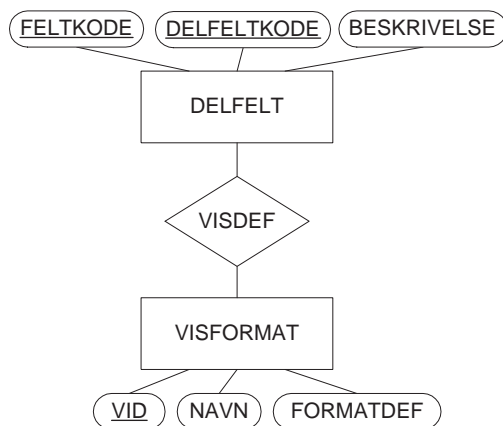
Søgekode	Beskrivelse
bi	Indholdsbeskrivende brødtekst
bt	Brødtekst
cl	Klassifikation
em	Emneord
fl	Forlagets navn
fo	Forfatter
ix	LIX
kk	Katalogkode
ma	Materialetype
no	Noter
nr	Numre
pu	Forlagets hjemsted
rt	Relationstitel, periodica
ti	Titel
vo	Vokabularium
ww	URL
år	Udgivelsesår

Tabel 4.3: Liste med de søgekoder, der er valgt for prototypen.

4.2.3 Visningsformat

Informanterne var enige om, at visning af fundne objekter skulle være fleksibel, primært ved at der skulle være mulighed for at vælge mellem forskellige visningsformater. Det er derfor vigtigt, at den datamodel, der vælges, ikke udelukker muligheden for at opbygge forskellige visningsformater. Det er ikke muligt at forudsige, hvilke formater der vil være behov for, så alle muligheder skal bibeholdes i modellen, hvilket opnås ved at fastholde

danMARC2-formatets høje detaljeringsgrad i datamodellen.



Figur 4.8: Datamodel for repræsentation visningsformater.

Figur 4.8 viser datamodellen til repræsentation af visningsformater, hvor felter og delfelter knyttes til formatet gennem relationen VISDEF mellem DELFELT og VISFORMAT.

Tabel 4.4 viser et eksempel på data fra felt 100, ”*Personnavn som opstillingselement*” for objektet med OID = 6826193. De tilknyttede delfelter *a*, *h* og *c* er henholdsvis efternavn, fornavn og fødselsår.

OID	FELT	DELFELT	DATA
6826193	100	a	Sørensen
6826193	100	h	Villy
6826193	100	c	f. 1929

Tabel 4.4: Eksempel på data fra felt 100 for objektet med OID = 6826193.

Formatdefinitionen, *formatdef* i VISFORMAT, beskriver formatet ved at benytte informationen i VISDEF. Hvis formatdefinitionen benytter variable med formen \$ < feltkode > < delfeltkode >, vil definitionen

Forfatter: \$100a, \$100h (\$100c)

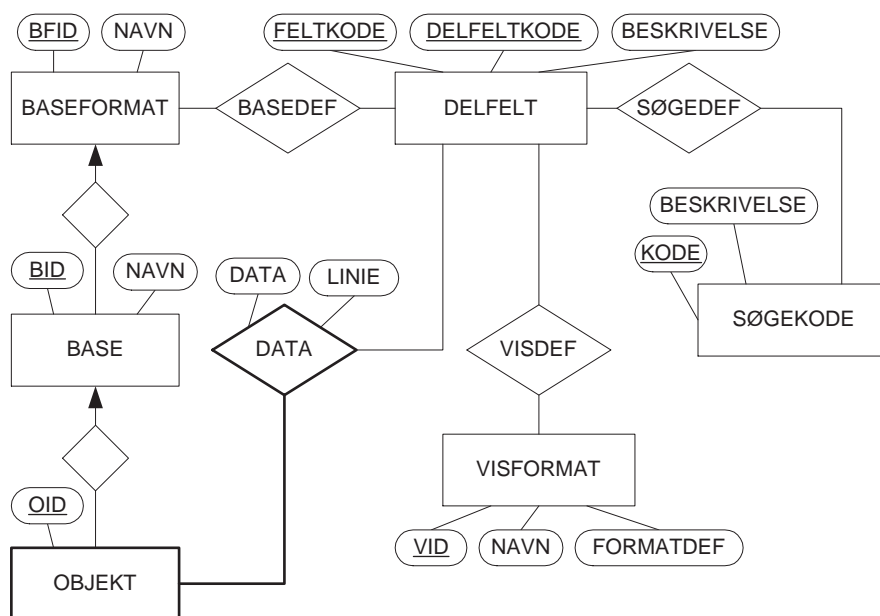
for indholdet i tabel 4.4 give følgende visning:

Forfatter: Sørensen, Villy (f. 1929)

I eksemplet benyttes felt og delfelt som variable, der kan indsættes i formatdefinitionen, og på denne måde opbygge et visningsformat. Det kræver

en syntaks for substituering af variabler og mulighed for at beskrive løkker for gentagende felter et cetera. Der kan dermed opnås en repræsentation, der er meget fleksibel, og som giver mulighed for at opbygge forskellige visningsformater. Der er naturligvis andre muligheder for at definere visningsformater, og valget af metode afhænger af de behov, brugerne har, og skal ikke fastlægges på nuværende tidspunkt. Jeg vil i dette projekt ikke komme yderligere ind på definition af visningsformater, og vælger for denne prototype at benytte, hvad man kunne kalde systemets standardvisningsformat, danMARC2-formatet.

4.2.4 Objekter, metainformation og visningsformat



Figur 4.9: Datamodel for repræsentation af objekter og metainformation.

I figur 4.9 vises datamodellen for repræsentation af objekter, metainformation og visningsformat. Et objekt tilhører en bestemt base og må derfor relatere til metainformationerne via tabellen BASE. Modellen fra figur 4.5b på side 29 er markeret, og entiteten DATA er her en relation mellem OBJEKT og DELFEKT, fordi informationen om felter og defelter er defineret i metainformationen. For at kunne vise objekterne i det oprindelige danMARC2-format tilføjes en positionering, attributten *linie* i DATA, der svarer til den rækkefølge, felterne er tilknyttet objekterne.

Datamodellen i figur 4.9 er designet til visning af objekter og kan ikke benyttes til effektiv søgning, fordi søgning i denne model ville betyde, at relationen DATA skulle benyttes. Alle relationerne i DATA skal gennemløbes for

at afgøre, om en given term matcher en eller flere af objekterne, hvilket er en langsom proces, der har et tidsforbrug svarende til strengsammenligning på alle data i samtlige delfelter for samtlige felter i samtlige objekter. Derfor må der tilføjes en datamodel, der understøtter effektiv søgning. Dette kaldes ofte indeksering.

4.2.5 Indeksering

Indekseringsprocessen er grundlaget for søgesystemets effektivitet og derfor af stor betydning, ikke mindst fordi et væsentligt succeskriterium for informanterne var søgesystemets svartider. Målet med indekseringsprocessen er at skabe beskrivelser af objekterne, der kan understøtte effektiv søgning. En ofte anvendt indekseringsmetode kaldes invertering, der vender problemstillingen om. I stedet for at gennemse alle objekterne efter en bestemt term, gennemses en liste af mulige termer for at finde de objekter, der har termen tilknyttet.

Objekt	Termer	Term	Objekter
O_1	A,B,C	A	O_1, O_3
O_2	C	B	O_1
O_3	A, C	C	O_1, O_2, O_3

(a)
(b)

Tabel 4.5: To tabeller, der viser hvordan indekseringen opbygger en liste med termer, der refererer til de objekter, hvori de optræder.

I tabel 4.5 vises et eksempel på, hvordan der ud fra objekter (tabel 4.5a) og de termer, de har tilknyttet, kan etableres en invertering (tabel 4.5b), hvor termerne relaterer til objekter. Det indeks, der etableres ved inverteringen, indeholder kun de enkelte termer en gang og kan derfor optimeres for søgning.

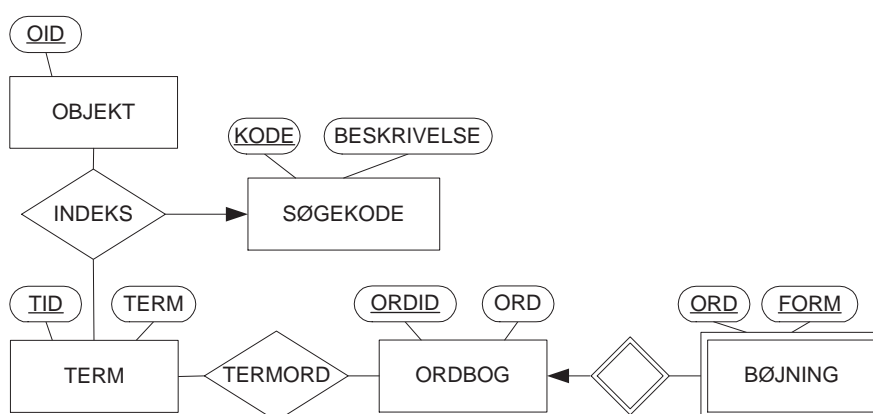
Normalisering af termerne er en måde, hvorved søgning i den inverterede liste af termer kan optimeres. Dette kan være ved at konvertere alle termer til enten store eller små bogstaver eller gennem behandlingen af specialtegn. For eksempel vil en konvertering af specialtegnet "-" til mellemrum samt en konvertering til små bogstaver betyde, at termen "*Kragh-Jacobsen*" bliver til termerne "*krag*" og "*jacobsen*". Dette vil betyde, at uanset hvordan bogstavkombinationen "*Kragh-Jacobsen*" eller "*Kragh Jacobsen*" optræder i objekterne, vil de i den inverterede termliste resultere i to termer. Foretages samme normalisering på forespørgslerne, vil det betyde, at det er underordnet om brugerne indtaster "*Kragh-Jacobsen*" eller "*KrAGh JacoBsen*", fordi de konverteres til de samme to termer. Dette er endnu et eksempel på at søgesystemet bliver mere tolerant overfor indtastning af forespørgsler.

En anden mulighed for normalisering er konvertering af ords forskellige bøjningsformer til en bestemt form, således at uanset, om der i objektet for eksempel står *bil*, *biler*, *bilerne*, *bilens*, refereres kun til for eksempel *bil*. Denne normalisering kan også foretages på forespørgslerne og derfor reducere fejl, der er afledt af indtastning af en anden bøjningsform end den, der benyttes i objekterne.

For at kunne foretage denne konvertering mellem ords bøjningsformer må der tilføjes leksikal viden, der kan benyttes til at afgøre, om to ord er forskellige bøjningsformer af det samme ord. Dette kan opnås ved at inddrage en ordbog med en form, svarende til eksemplet i tabel 4.6, hvor ordene er foldet ud i de bøjningsformer, der er kendte. Målet er ikke at kunne håndtere alle ord, men at de ord, der kendes, kan behandles.

ORDID	Ord	Form
5754	bil	0
5754	bilen	1
5754	biler	2
5754	bilerne	3

Tabel 4.6: Eksempel på, hvordan en ordbog kan udfoldes.



Figur 4.10: Datamodel over repræsentation af indeksering.

Datamodellen i figur 4.10 viser repræsentation af indeksering og ordbog. Relationen INDEKS mellem TERM og OBJEKT relaterer også til SØGEKODE for at give mulighed for at afgrænse søgningen til en bestemt delmængde af objekterne. Relationen TERMORD mellem TERM og ORDBOG er en mapping mellem ord i ordbogen og termer.

Indekseringen af objekterne resulterer i beskrivelser af objekterne. En beskrivelse er den mængde af termer, der tilknyttes objektet ved indekseringen.

4.2.6 Viden

Den type af viden, der skal kunne tilknyttes søgesystemet, kan opdeles i to grupper: internt afledt og eksternt tilknyttet viden. Den internt afledte viden er viden, der dannes på baggrund af objekterne i søgesystemet, mens

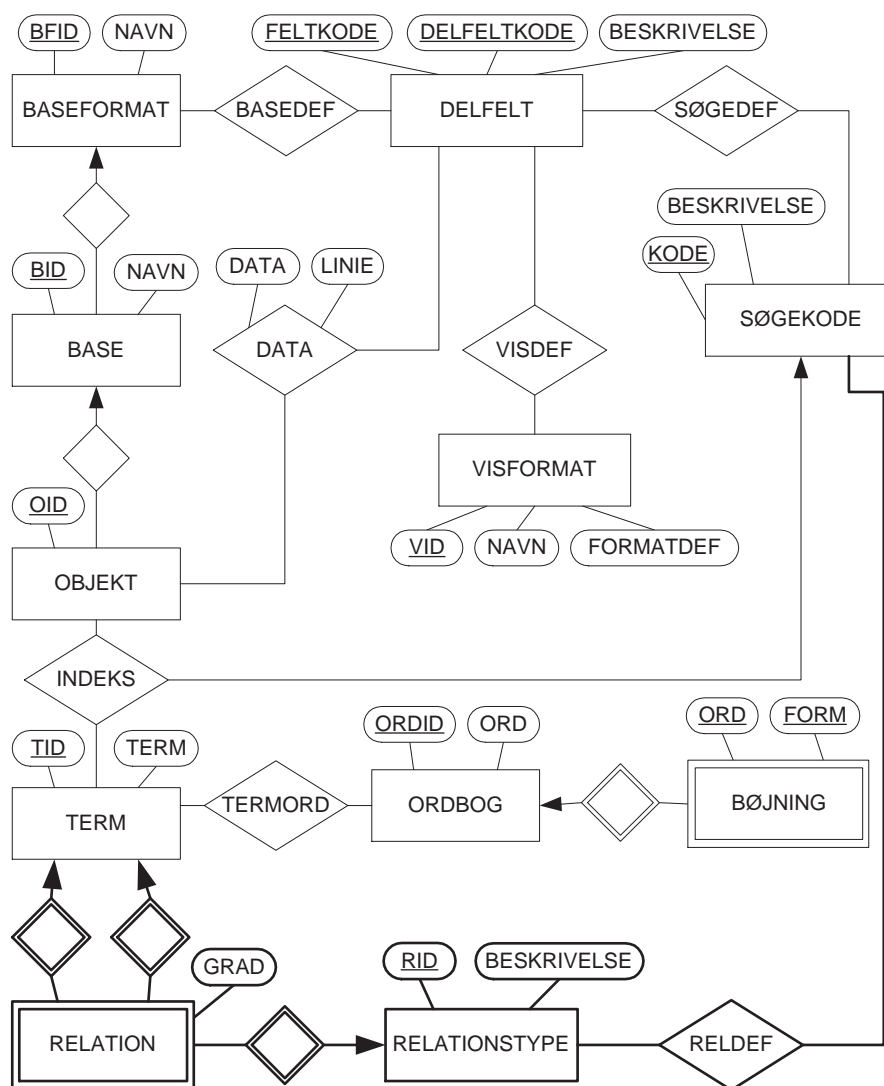
den eksternt tilknyttede viden ikke har en direkte relation til objekterne i systemet, men er almen viden. Et eksempel på internt afledt viden er associationer dannet på baggrund af termers ko-forekomster i objekter eller viden afledt af systemets brug, for eksempel www.amazon.com's statistik over, hvilke bøger der er købt sammen. Dette er også eksempel på to forskellige relationer, term-term relationer og objekt-objekt relationer. I denne prototype er det term-term relationerne, der behandles. Den eksternt tilknyttede viden kan være forskellige semantiske eller leksikale relationer, for eksempel ontologi eller synonymi. En anden måde at differentiere mellem de to typer viden er, at den internt afledte viden er statistisk udledt, mens den eksternt tilknyttede viden er skabt intellektuelt⁷.

Formålet med inddragelse af viden er at blive i stand til at ekspandere forespørgslerne. Forespørgslen $\{A, B\}$ skal for eksempel kunne ekspanderes til $\{\{A, A_1, A_2\}, \{B, B_1\}\}$, hvor A_1 og A_2 har en relation til A , og B_1 en relation til B . Dermed er det ikke kun objekter, der opfylder A og B , men også objekter, der for eksempel opfylder A_1 og B , der kan medtages i svaret.

Det skal ved udvidelse af forespørgslerne med relaterede termer være muligt at adskille den oprindelige term fra de ekspanderede, således at objekter, der matcher den oprindelige term, kan vægtes og rangeres højest. Der bør også kunne differentieres mellem de relaterede termer, således at søgesystemet kan kontrollere ekspansionen og udfra graden af sammenhæng vælge, hvilke relationer der skal benyttes. Uden denne mulighed for differentiering ville det enten betyde, at alle relationer skulle være af en meget høj kvalitet, eller at der ville opstå støj, fordi mindre gode eller misvisende relationer kan betyde, at der i svaret vil optræde objekter, der er umulige at relatere til udgangspunktet. En relation tilknyttes derfor en grad, der vægter den relationelle sammenhæng.

I figur 4.11 vises den samlede datamodel for hele søgesystemet, hvor den del der omhandler repræsentation af viden er markeret. Entiteten RELATION repræsenterer viden med relationer til TERM og RELATIONSTYPE. Entiteten RELATION er en svag entitet, idet den således bliver identificeret igennem den sammenhæng til de termer den tilknytter, samt til relationstypen. RELATIONSTYPE relaterer til SØGEKODE via relationen RELDEF. Relationen mellem RELATIONSTYPE og SØGEKODE definerer afhængigheden mellem viden og søgekoder, således at der kan indføres begrænsninger for, på hvilke dele af objekterne viden kan inddrages. For eksempel kunne der for en bestemt viden defineres, at den kun kunne anvendes på emneord.

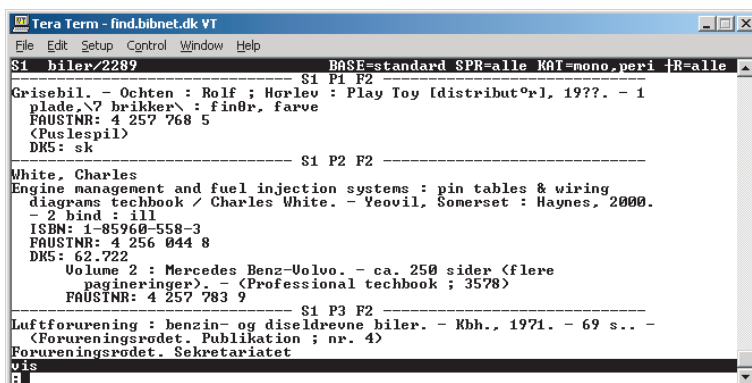
⁷Det kan naturligvis diskuteres om det er berettiget at anvende betegnelsen "viden" om statistisk udledte informationer. Det skal dog bemærkes, at dette nu er gængs sprogbrug.



Figur 4.11: E/R diagram over hele repræsentationen.

4.3 Brugergrænseflade

Der var blandt informanterne enighed om, at de foretrak den tegnbaserede grænseflade fremfor den web-baserede. Den generelle holdning var, at de tegnbaserede grænseflader er hurtigere, lettere at bruge og har flere tilpasningsmuligheder. For eksempel kan man vælge mellem mange forskellige visningsformater og benytte søgesæt i forespørgselsdefinitionen. Et andet argument var, at de tegnbaserede grænseflader er bedre integreret med de administrative systemer, hvilket betyder at der fra samme grænseflade er adgang til flere informationer.



Figur 4.12: DBC's tegn-baseret grænseflade til DanBib.

4.3.1 Tegnbaserede grænseflader

Tegnbaserede grænseflader benyttes via en terminal, der forbinder til den server, hvorpå søgesystemet tilbydes. I figur 4.12 vises DanBib's tegnbaserede grænseflade. Den nederste linie (der begynder med et kolon) er kommandolinien, hvor kommandoerne til søgesystemet kan indtastes. Den sorte linie lige over viser sidste kommando, her kommandoen *vis*. Al kommunikation med grænsefladen sker gennem kommandoer, og det er derfor ikke muligt at benytte musen eller piletasterne til at navigere med. Ønsker man for eksempel at blade en side frem, benyttes kommandoen *m* (for mere). Den øverste linie viser blandt andet, hvilket søgesæt der vises, her søgesæt 1 (*s1*), forespørgslen og antal fundne objekter. Et søgesæt er den mængde objekter, der er fundet ved en forespørgsel. Søgesættene nummereres og kan derfor benyttes i nye forespørgsler. Søgesæt 1 i figur 4.12 indeholder de 2289 objekter, der matcher forespørgslen "biler". Vil man derefter søge efter objekter, der matcher forespørgslen "biler og rejser", kan det udtrykkes som "s1 og rejser".

Common Command Language

Kommandosproget, der benyttes i DanBib, hedder Common Command Language (CCL). CCL er en international standard, der definerer kommandoer og syntaksen for, hvordan disse benyttes. Denne standard er konverteret til dansk i [Dansk standardiseringsråd, 1990], og det er denne danske udgave af CCL, der benyttes i de danske bibliografiske søgesystemer, der understøtter CCL. Udover styrekommandoer til grænsefladen er CCL også et forespørgselsprog med mulighed for at udtrykke komplekse forespørgsler.

Forespørgsler i CCL har følgende generelle struktur [Dansk standardiseringsråd, 1990, side 9]

$$\langle \text{søgeargument} \rangle = \langle \text{søgeelement} \rangle \langle \text{operator} \rangle \langle \text{søgeelement} \rangle \quad (4.1)$$

hvor *søgeelementerne* kan være en søgeterm, som kan være maskeret, et intervaludtryk eller identifikatorer for tidligere forespørgsler. Operatorerne er de logiske operatoren *AND*, *OR* og *NOT*, samt intervaloperatorer og en nærhedsoperator. Intervaloperatorerne er de matematiske operatoren $<$, $>$, $<=$, et cetera, og operatoren $-$ (*TO*), for eksempel $\text{år} >= 1990$ eller $\text{år} = 1990-1995$. Nærhedsoperatoren $\#$ ⁸ benyttes til at bestemme, hvor stor afstand der må være mellem to termer. For eksempel vil udtrykket *rejser # børn* betyde, at der højst må være en term mellem *børn* og *rejser*.

Forespørgslen *"børn rejser"* evalueres som strengen *"børn rejser"*. Ønskes der en søgning på de enkelte termer, skal der benyttes operatoren imellem, for eksempel *"børn og rejser"*. Skal der søges på strengen *"børn og rejser"* skal *og* indesluttet mellem restituerings tegn, *"børn "og" rejser"*.

Til tegnmaskering kan der anvendes tre forskellige maskeringstegn; $!$ nøjagtig et tegn, $\#$ nul eller et tegn og $?$ nul eller flere. For eksempel vil forespørgslen *e#mail* matche både *email* og *e-mail*.

Det er også muligt at kontrollere, i hvilken del af objekterne der skal søges, med henholdsvis præfiks og suffiks. Forespørgslen *ti=børn rejser* benytter søgekoden *ti* til at afgrænse søgningen til titler, eller *børn rejser/ti,em*, der definerer, at der skal søge i titler og/eller emneord.

4.3.2 Web-baserede grænseflader

I figur 4.13 vises DBC's web-baserede brugergrænseflade *DanWeb*, hvor nogle af CCL's muligheder er forsøgt inkluderet i den grafiske grænseflade. For eksempel kan der vælges materiale type, hvilket svarer til at benytte præfikset *ma* i CCL. Informanternes holdning til *DanWeb* var entydig, udover at være fantastisk langsom, er *DanWeb* vanskelig at bruge, fordi det blandt andet tager meget længere tid at udtrykke forespørgsler, end i CCL. Den mangler mulighed for at benytte søgesæt til definition af forespørgsler,

Figur 4.13: DBC's web-baserede grænseflade til DanBib.

⁸Bemærk at $\#$ også benyttes til tegnmaskering, men når den optræder som nærhedsoperator, skal der være mellemrum før og efter.

og det er ikke muligt at vælge forskellige visningsformater.

Figur 4.14 viser en helt ny brugergrænseflade til DanBib, *bibliotek.dk*. Den udemærker sig i forhold til *DanWeb* først og fremmest ved at være væsentlig hurtigere, og grænsefladen er blevet mere simpel, hvor det kun er nogle få elementer fra CCL, der er inkluderet. Dette skyldes primært at målgruppen for *bibliotek.dk* er den brede befolkning, i modsætning til *DanWeb*, der har de professionelle brugere som målgruppe. Dette forklarer dog ikke, hvorfor *DanWeb* bruger 5-6 gange mere tid til at evaluere samme forespørgsel som *bibliotek.dk*.



Figur 4.14: Bibliotek.dk er en nyudviklet grænseflade til søgning i DanBib.

Begge disse web-baserede brugergrænseflader har muligheden for at vælge en kommando-grænseflade, der understøtter forespørgselsdelen i CCL. Alle informanter var enige om, at *DanWeb* er så langsom, at det uanset grænseflade er usandsynligt, at professionelle søgere vil benytte den. *Bibliotek.dk* har derimod fornuftige svartider, og en del af informanterne benyttede den, primært når de søgte sammen med lånerne. Den er dog så ny, at informanterne ikke havde et reelt grundlag for at vurdere den.

4.3.3 Prototypens grænseflade

Formålet med denne prototype er at undersøge, om ekspandering af forespørgsler kan understøtte den søgning, bibliotekarer kalder *quick and dirty*. Den brugergrænseflade, der er behov for, skal derfor primært understøtte dette formål, med udgangspunkt i hvilke generelle krav informanterne havde.

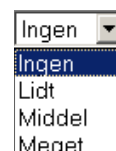
Et væsentligt krav fra informanterne var at grænsefladen understøttede CCL, og at den ikke, som for eksempel *DanWeb*, har en masse forskellige CCL-parametre, der kontrolleres grafisk. Herudover skulle grænsefladen rumme mulighed for at kontrollere ekspansionen af forespørgslerne, ligesom det skulle være muligt at vælge ekspansion til og fra.

I figur 4.15 vises den del af prototypens brugergrænseflade, der benyttes til at definere forespørgsler. Det centrale er et indtastningsfelt, der understøtter CCL, og knappen *Find* til eksekvering af forespørgslen. Herunder er der fire forskellige valgmuligheder; *Ekspansion*, *Evaluering*, *Visning* og *Antal*.



Figur 4.15: Udsnit af prototypens grænseflade, hvor forespørgslerne defineres.

I *Ekspansion* kan der vælges, hvor meget der skal ekspanderes. I figur 4.16 ses en udfoldning af valgmulighederne. *Lidt* betyder, at der kun skal ekspanderes en lille smule, hvilket vil sige, at de relationer, der inddrages, skal have en høj grad af sammenhæng. *Meget* betyder, at alle relationer med en grad af sammenhæng, der er større end 0, inddrages.



Figur 4.16: Valgmuligheder for ekspansion.

I *Evaluering* kan man bestemme, hvilken evalueringsmetode der skal benyttes. Mulighederne er henholdsvis *Boolesk* og *Fuzzy*, som det ses i figur 4.17. Den booleske evaluering inddrager ikke ekspansion, hvilket betyder at valget i *Ekspansion* ikke har indflydelse på den booleske evaluering.



Figur 4.17: Valgmuligheder for evaluering.

Visning bestemmer, hvilket visningsformat objekterne skal vises i. Der kan i denne prototype ikke vælges mellem forskellige visningsformater, idet det kun er DanMARC2-formatet, der understøttes. *Antal* bestemmer, hvor mange objekter der skal vises på en side.

Figur 4.18 viser resultatet af en forespørgsel, hvor der øverst gives oplysninger om forespørgsel, antal fundne objekter og tidsforbrug. Tidsforbruget er opdelt, så det viser forbruget i de forskellige dele af evalueringen. Herefter vises de fundne objekter med nummerering, grad og objektidentifikation.

4.3.4 Associationsnavigator

Der var blandt informanterne enighed om, at et associationsnet baseret på termers ko-forekomster i objekterne var interessant, og at det kunne benyttes som hjælp til definition af forespørgsler på linie med ordbøger, begrebshierarkier og andet.

Den viden, der inddrages i prototypen, er relationer mellem termer, og hvis det skal være muligt for brugerne at benytte denne viden til definition af forespørgsler, er det nødvendigt med en grænseflade, hvor der kan navigeres

GISS
Indtast forespørgsel

em=(ungdom narkomani)

Ekspansion: Meget | Evaluering: Fuzzy | Visning: Default | Antal: 10

Forespørgsel: em=(ungdom narkomani)
Antal objekter: 21717
Søgetid: 0.72 (DB select: 0.01 Fetch: 0.44 Aggregering: 0.15 Sortering: 0.1)

1. GRAD: 0.599282 ObjektID: 735871

001 00 *a8798399403
008 00 *a1991 *l dan
009 00 *g:xx
021 00 *a87-983994-0-3
245 00 *a Unges forbrug af rusmidler i Nykøbing F.
260 00 *a[Nykøbing F.]*b[SSP-gruppen i Nykøbing F.]*c[1991]*grevirering ... SSP-sekretariatet
512 00 *a Rapporten er udarbejdet af en arbejdsgruppe under SSP-gruppen
652 00 *p66.88 *p66.83 *p30.1664
661 00 *aalkoholisme *aDanmark *anarkomani *arygning *astatistiske data *aunge
710 00 *aNykøbing Falster *cSSP-gruppen
910 00 *aSSP-Gruppen i Nykøbing F.
m01 00 *ate

Figur 4.18: Et eksempel på resultatet af en forespørgsel i prototypens grænseflade.

rundt mellem termernes relationer.

I figur 4.19 vises et eksempel på associationsnavigering for termen *børneforskning*. Indtastningsfeltet øverst benyttes til at søge efter termer. Findes

Associationsnavigator

børneforskning

børneforskning

0.8332	daginstitutionsforskning	0.887	forskning
0.7999	barndomsforskning	0.784	barn
0.4999	norsk senter for barneforskning	0.532	børneopdragelse
0.4999	prout	0.5317	opdragelse
0.4999	theorizing childhood	0.1859	barndom
0.3333	olsén	0.1627	børnekultur
0.3333	småbørnsinitiativet	0.1395	småbørnsforskning
0.25	jencks	0.1163	daginstitutionsforskning
0.2069	småbørnsforskning	0.1153	daginstitution
0.15	ungdomsforskning		

Figur 4.19: Et eksempel på associationsnavigering for termen *børneforskning*.

der mere end en term, der matcher det indtastede, gives en liste med mulighed for at vælge den term, der ønskes vist. Under indtastningsfeltet vises relationerne mellem termer. Den valgte term vises øverst, og relationerne vises i to kolonner sorteret faldende efter grad. Den venstre kolonne viser, hvilke termer der kan ekspanderes med den viste term, og den højre kolonne, hvilke

termer den valgte term kan ekspanderes med. For eksempel ville en søgning på termen *daginstitutionsforskning* kunne ekspanderes med *børneforskning* til graden 0.8332, "(*daginstitutionsforskning*, 1),(*børneforskning*, 0.8332)", hvilket betyder at *børneforskning* beskriver *daginstitutionsforskning* med graden 0.8332. Den anden vej kan *børneforskning* ekspanderes med alle termerne i højre kolonne "(*børneforskning*, 1),(*forskning*, 0.887), . . . , (*daginstitution*, 0.1153)", hvor graderne beskriver, i hvor høj grad de ekspanderede termer beskriver *børneforskning*.

4.4 Associationsnet

Den viden, jeg ønsker at inddrage i denne prototype, er et associationsnet, der er afledt på baggrund af termers ko-forekomster i objekter. Det er altså den type viden, der beskrives som internt afledt viden, i afsnit 4.2.6 på side 35. Associationer, der udledes fra objekterne i søgesystemet, har den specielle egenskab, at de beskriver de sammenhænge, der er gyldige for den kombination af objekter, der er repræsenteret i søgesystemet. I modsætning til almene relationer, der forsøger at beskrive den universelle virkelighed, er målet med denne type relationer ikke at forsøge at forklare verden, men at beskrive søgesystemets datagrundlag.

Associationsnettet kan benyttes i forespørgselsevalueringen til ekspansion af forespørgslerne, og hvis nettet indeholder alt for meget støj i form af meningsløse relationer, vil det betyde, at systemet fremkommer med mærkelige og helt uforståelige svar. Det er derfor nødvendigt at have kontrol over, hvilke termer der benyttes. Projektet "*Fleksibel søgning i DanBib*" [Andreasen, 1998], [Forsberg *et al.*, 1999], benyttede emneord til etableringen af associationsnettet og fandt, at dette gav et net med en høj kvalitet. Emneordene er manuelt tildelt objekterne i DanBib, hvilket betyder, at en indeksør har overvejet, hvilke emneord der skal tilknyttes for at beskrive de enkelte objekter. Det er den primære årsag til den høje kvalitet.

I afsnit 4.2.2 på side 31 oprettede jeg gruppen (søgekoden) "*vo*" med det formål at definere den del af objekterne, der kan benyttes ved etablering af associationsnettet. Denne gruppe definerer et vokabularium med en delmængde af objekternes emneord. Selektionen af felter/delfelter i forbindelse med definitionen af vokabulariet har handlet om at udvælge de meningsbærende dele fra emneordene. Et eksempel er felt 600, "*Personnavn som emneord*", delfeltet *c*, *fødselsår*, med formen *f. årstal*, som jeg ikke mente var relevant i forbindelse med etablering af associationsnettet, og derfor ikke er medtaget i vokabulariet.

I tabel 4.7 vises, hvordan sammenhængen mellem søgekoder og viden beskri-

RELATIONSTYPE		RELDEF		SØGEKODE	
RID	Beskrivelse	RID	KODE	Kode	Beskrivelse
1	Associationsnet	1	vo	vo	Vokabularium

Tabel 4.7: Et eksempel på hvordan vidensrepræsentationen benyttes til af definere hvordan sammenhængen er mellem viden og søgekoder.

ves i datamodellen, og på den måde kan beskrive, hvilke dele af objekterne der skal benyttes til etablering af associationsnettet.

Når to termer er tilknyttet det samme objekt, har de det tilfælles, at de sammen er med til at beskrive objektets indhold. Der opstår en relation mellem termene, netop fordi de er tilknyttet samme objekt, og denne relation forstærkes, hvis de optræder sammen i flere objekter. Hvis alle objekter, der har den ene term tilknyttet, også har den anden og omvendt, er de i forhold til søgning synonyme. Der vil ikke være forskel på, om der søges på den ene eller den anden term. Det vil være den samme mængde objekter, der findes. Relationen, der opstår, er en association. Den beskriver termernes indbyrdes evne til at beskrive hinanden.

Antallet af ko-forekomster beskriver graden af sammenhæng eller associationsgraden. Hvis funktionen $p(term_i)$ returnerer den mængde objekter, der har $term_i$ tilknyttet, og $|p(term_i)|$ antallet af objekter, vil associationsgraden mellem $term_i$ og $term_j$ kunne beregnes på følgende måde

$$associationsgrad = ass(term_i, term_j) = \frac{|p(term_i) \cap p(term_j)|}{|p(term_i)|} \quad (4.2)$$

og bestemme, i hvor høj grad $term_i$ associerer til $term_j$. Hvis $term_i$ er tilknyttet alle de objekter, $term_j$ er tilknyttet, bliver associationsgraden 1, og hvis de ikke optræder sammen i nogle objekter, bliver den 0. Associationsgraden mellem termer vil beskrive graden af sammenhæng i intervallet $[0,1]$. Associationsgraden 0 betyder, at der ingen sammenhæng er, og 1 betyder, at de er synonyme. Det er dog vanskeligt at forstille sig, at det på baggrund af statistik giver mening at udlede 1-1 synonymer. Målet med ekspansionen er at inddrage ”noget der ligner” og dermed opbløde evalueringen, men altid at kunne præsentere de objekter, der opfylder den oprindelige forespørgsel som det bedste svar. Der opstilles derfor følgende regulerende funktion for associationerne

$$ass(term_i, term_j) = \begin{cases} 1 & \text{hvis } term_i = term_j, \\ \frac{|p(term_i) \cap p(term_j)|}{|p(term_i)|+1} & \text{hvis } term_i \neq term_j \end{cases} \quad (4.3)$$

som fjerner muligheden for, at en relation kan opnå samme vægt som termen, den relaterer til, fordi

$$|p(term_i)| + 1$$

i nævneren sikrer, at associationsgraden mellem to forskellige termer aldrig kan blive 1.

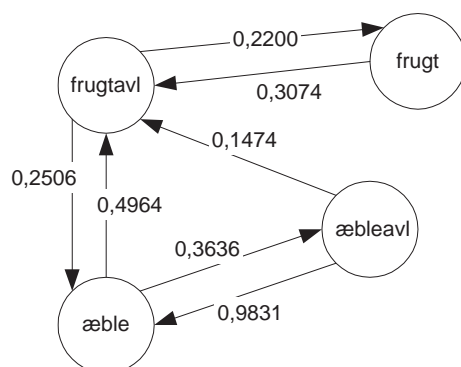
Hvis $term_1$ er tilknyttet 25 objekter, og $term_2$ er tilknyttet 10, og de optræder sammen i 7 objekter, vil associationerne mellem $term_1$ og $term_2$ med denne model være

$$ass(term_1, term_2) = \frac{|p(term_1) \cap p(term_2)|}{|p(term_1)| + 1} = \frac{7}{25 + 1} = 0,2692 \quad (4.4)$$

der beskriver, at $term_1$ associerer til $term_2$ ($term_1 \rightarrow term_2$) med graden 0,2692, og

$$ass(term_2, term_1) = \frac{|p(term_2) \cap p(term_1)|}{|p(term_2)| + 1} = \frac{7}{10 + 1} = 0,6364 \quad (4.5)$$

at $term_2 \rightarrow term_1$ med graden 0,6364.



Figur 4.20: Udsnit af et associationsnet illustreret som en orienteret graf.

Associationerne kan illustreres som en orienteret graf. Figur 4.20 viser et eksempel på et udsnit af et associationsnet.

Associationerne, der dannes med denne model, inddrager kun den ene terms objektfrekvens⁹, hvilket er problematisk, fordi en association til en højfrekvent term ikke adskiller sig fra en association til en mindre frekvent term. I klassiske information retrieval (IR) modeller benyttes den omvendte dokumentfrekvens (inverted document frequency) [Salton, 1988] til at beskrive, hvor god en term er til at skelne mellem objekter. En term, der optræder

⁹Antallet af objekter, hvor termen er tilknyttet.

i alle objekter, vil ikke skelne de enkelte objekter fra hinanden, på samme måde som en term, der kun forekommer i få objekter. Den omvendte dokumentfrekvens kan beregnes med [Salton and McGill, 1983]

$$\log_2 \frac{n}{|p(\text{term})|} + 1 \quad (4.6)$$

hvor n er det totale antal objekter. Modellen skal modificeres så den benytter den omvendte dokumentfrekvens til at reducere associationsgraden, når der associeres til højfrequente termer. Modellen tilføjes den omvendte dokumentfrekvens og en konstant k , der kan bruges til at kontrollere effekten af denne

$$\text{ass}(\text{term}_i, \text{term}_j) = \frac{|p(\text{term}_i) \cap p(\text{term}_j)|}{|p(\text{term}_i)|} \left(1 - k \left(\frac{|p(\text{term}_j)|}{OF_{max}} \right) \right) \quad (4.7)$$

OF_{max} er den maksimale objektfrekvens, der sikrer at

$$\frac{|p(\text{term}_j)|}{OF_{max}} \leq 1 \text{ for alle termer.}$$

Reduktionen af associationsgraden kan maksimalt blive $1 - k$. Hvis $OF_{max} = 30$ og $k = 0,75$, vil eksemplerne i (4.4) og (4.5) med denne model give følgende grader

$$\text{ass}(\text{term}_1, \text{term}_2) = \frac{7}{25 + 1} \left(1 - 0,75 \left(\frac{10}{30} \right) \right) = 0,2019 \quad (0,2692) \quad (4.8)$$

og

$$\text{ass}(\text{term}_2, \text{term}_1) = \frac{7}{10 + 1} \left(1 - 0,75 \left(\frac{25}{30} \right) \right) = 0,2386 \quad (0,6364) \quad (4.9)$$

med den tidligere grad i parentes. I (4.8) reduceres graden med $0,2692 - 0,2019 = 0,0673$, og i (4.9) med $0,6364 - 0,2386 = 0,3978$, fordi term_1 forekommer i 83% af objekterne, mens term_2 kun forekommer i 33% af objekterne. Den ønskede effekt er dermed opnået, og denne model benyttes i søgesystemet til etablering af associationsnettet.

4.5 Forespørgselsevaluering

I indekseringen dannes der beskrivelser af objekterne gennem inverteringen i form af den mængde termer, der er tilknyttet objekterne. Ved at omforme forespørgslerne til tilsvarende beskrivelser, altså en mængde af termer, vil evalueringen af forespørgsler være en sammenligning af beskrivelser. Denne

sammenligning skal bestemme, i hvor høj grad objekternes beskrivelser er identiske med forespørgslens.

Søgesystemer, der anvender boolesk logik, kan afgøre, om objektets beskrivelse opfylder forespørgslen eller ej. For eksempel vil forespørgslen (A) kun blive opfyldt af alle de objekter, der har A som element i beskrivelsen. Med funktionen $p(x)$, der returnerer de objekter, der har term x tilknyttet, vil dette kunne udtrykkes som $p(A)$. Forespørgslerne med operatorerne *og*, *eller* og *ikke* vil kunne evalueres som henholdsvis

$$\begin{aligned}(A \text{ og } B) &= p(A) \cap p(B) \\(A \text{ eller } B) &= p(A) \cup p(B) \\(A \text{ og ikke } B) &= p(A) - p(B)\end{aligned}$$

hvor resultatet kan antage værdierne falsk, sand eller 0, 1.

Hvis forespørgslerne ekspanderes, vil der være behov for at kunne bestemme graden af lighed på et mere generelt niveau, således at et objekt kan opfylde en forespørgsel til en hvis grad. Denne grad af lighed kaldes ofte similaritet. Dette er muligt med fuzzy logik, hvor sandhedsværdierne ofte beskrives med intervallet $[0, 1]$, hvor 0 er falsk og 1 er sandt, og alt derimellem er sandt til en hvis grad.

4.5.1 Fuzzy-mængder

I klassiske mængder kan man skelne mellem medlemmer og ikke-medlemmer. I fuzzy-mængder generaliseres dette således, at et element kan have et delvist tilhørsforhold til mængder. Dette udtrykkes ofte med medlemskabsfunktioner [Klir and Yuan, 1995]

$$m_A : X \rightarrow [0, 1]$$

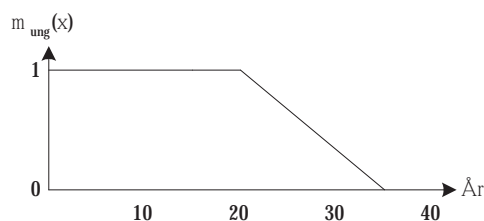
der beskriver graden af medlemskab for elementerne i mængden. Dette er en generalisering, hvor den klassiske mængde kan beskrives som et specialtilfælde med medlemskabsfunktionen

$$m_A(x) = \begin{cases} 1 & \text{for } x \in A \\ 0 & \text{for } x \notin A \end{cases} \quad (4.10)$$

Den generalisering, der indføres med fuzzy-mængder, åbner mulighed for at beskrive begreber på en anden måde, end de klassiske mængder tillader. Et eksempel kunne være begrebet *ung*, hvor det ikke ville være rigtigt at definere, at man er ung, hvis man er 20 år gammel, men ikke hvis man er 21. Det er ikke realistisk at bestemme, hvornår man holder op med at være ung, som et skæl mellem to år. Det kræver en blødere overgang, som netop kan beskrives med fuzzy-mængder. Følgende medlemskabsfunktion

$$m_{ung}(x) = \begin{cases} 1 & \text{hvis } x \leq 20 \\ (35 - x)/15 & \text{hvis } 20 < x < 35 \\ 0 & \text{hvis } x \geq 35 \end{cases} \quad (4.11)$$

definerer, at man frem til det fyldte 20 år er ung, mellem 20 og 34 er ung til en hvis grad og når man er mere end 34, holder helt op med at være ung. Medlemskabsfunktion $m_{ung}(x)$ illustreres i figur 4.21.



Figur 4.21: Illustration af begrebet ung.

4.5.2 Order Weighted Average aggregering

Forespørgsler kan være sammensat af flere kriterier, så der er behov for at kunne samle graden af opfyldelse for flere kriterier, hvilket kaldes aggregering. I den booleske logik sammensættes kriterier med operatorer og aggregeringen resulterer i sandhedsværdierne 1 eller 0. En aggregering i forbindelse med fuzzy-mængder skal resultere i værdier i intervallet $[0, 1]$.

Order Weighted Average aggregering (OWA) [Yager, 1988, side 183-190] er en aggregeringsfunktion, der ved hjælp af ordensvægte kan styre aggregeringen mellem *og* og *eller*. En forespørgsel med n kriterier tilknyttes n ordensvægte $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)$ for hvilke det gælder at

$$1 = \sum_{i=1}^n \alpha_i, \text{ hvor } \alpha_i \in [0, 1] \quad (4.12)$$

Princippet er, at kriterierne i forespørgslen sorteres faldende i forhold til sandhedsværdierne og vægtes med henholdsvis $\alpha_1, \alpha_2, \dots, \alpha_n$, således at kriteriet med højest grad vægtes med α_1 og så videre.

Lad objekterne O_1 og O_2 være beskrevet med $O_1 = \{term_1, term_2\}$ og $O_2 = \{term_2, term_3\}$ og forespørgslen $F = \{term_1, term_2\}$, således at similariteten kan beskrives som $(F, O_1) = (1, 1)$ og $(F, O_2) = (0, 1)$, fordi O_1 indeholder begge de termer som F indeholder, mens O_2 kun indeholder den ene. Herved vil følgende ordensvægte $\alpha = (1, 0)$, $\alpha = (0, 1)$ og $\alpha = (\frac{1}{2}, \frac{1}{2})$ betyde, at evalueringen vil være

$$\begin{aligned} owa_{\alpha=(1,0)}(F, O_1) &= (1 * 1 + 0 * 1) = 1 \\ owa_{\alpha=(1,0)}(F, O_2) &= (1 * 1 + 0 * 1) = 1 \end{aligned} \quad (4.13)$$

$$\begin{aligned} owa_{\alpha=(0,1)}(F, O_1) &= (0 * 1 + 1 * 1) = 1 \\ owa_{\alpha=(0,1)}(F, O_2) &= (0 * 1 + 0 * 0) = 0 \end{aligned} \quad (4.14)$$

$$\begin{aligned} owa_{\alpha=(\frac{1}{2},\frac{1}{2})}(F, O_1) &= (\frac{1}{2} * 1 + \frac{1}{2} * 1) = 1 \\ owa_{\alpha=(\frac{1}{2},\frac{1}{2})}(F, O_2) &= (\frac{1}{2} * 1 + \frac{1}{2} * 0) = 1/2 \end{aligned} \quad (4.15)$$

Funktionen $owa_{\alpha=(1,0)}$ i (4.13) beregner maksimum, eller hvad der svarer til *eller*. $owa_{\alpha=(0,1)}$ i (4.14) beregner minimum, som svarer til *og*. $owa_{\alpha=(\frac{1}{2},\frac{1}{2})}$ i (4.15) beregner gennemsnittet som vil svare til noget mellem *eller* og *og*. I prototypen har jeg valgt at benytte *gennemsnits*-aggregeringen, fordi den ikke er så restriktiv som *minimum*, men heller ikke tager alt med som *maksimum*, og derfor vil være et kvalificeret bud på en blødere og mere fleksibel evaluering.

4.5.3 Forespørgselsprog

I CCL er operatorerne de separatorer, der opdeler forespørgslerne i forskellige argumenter. For eksempel består CCL-forespørgslen *birthe arnbak og morgenstund* af to argumenter *birthe arnbak* og *morgenstund* separeret af operatoren *og*.

Evaluering med OWA eliminerer brugen af de booleske operatorer og erstatter dem med en fleksibel operator, der kan kaldes *samt* [Forsberg *et al.*, 1999]. Dette betyder, at det ikke er nødvendigt at eksplicitere operatorer i forespørgslen, hvis blot argumenterne i forespørgslen adskilles med en separator. I web-baserede applikationer er separatoren mellem argumenterne ofte mellemrum, der fortolkes som enten *og* eller *eller*.

Jeg valgte at ville afprøve et operatorløst forespørgselsprog. Dette krævede en redefinering af CCL, fordi operatorerne benyttes som separator mellem argumenterne. Jeg ville gerne benytte strategien fra de web-baserede applikationer med mellemrum som separator mellem argumenter. I DanBib er det muligt at søge på strenge, som for eksempel kan være en titel. Dette understøttes ikke i denne prototype. Det er derfor mere naturligt, at et mellemrum adskiller to termer, end at det er en del af en streng. For eksempel vil forespørgslen "*en lille bil*" i CCL fortolkes som strengen "*en lille bil*", mens jeg ønskede en fortolkning som tre termer, "*en*", "*lille*" og "*bil*". I figur 4.22 vises en definition af forespørgselsproget, der benyttes i prototypen. Det er en delmængde af CCL modificeret med hensyn til ovenstående valg.

Forespørgsel	::=	$\langle \text{ElementListe} \rangle$
ElementListe	::=	$\langle \text{Element} \rangle \mid \langle \text{ElementListe} \rangle \langle \text{Element} \rangle$
Element	::=	$\langle \text{PræfiksListe} \rangle = (\langle \text{TermListe} \rangle) \mid$ $\langle \text{PræfiksListe} \rangle = \langle \text{Term} \rangle \mid$ $\langle \text{TermListe} \rangle \mid \langle \text{ÅrPræfiks} \rangle$
PræfiksListe	::=	$(\langle \text{PræfiksListe} \rangle) \mid$ $\langle \text{Præfiks} \rangle \mid$ $\langle \text{PræfiksListe} \rangle \langle \text{Seperator} \rangle \langle \text{Præfiks} \rangle \mid$
Præfiks	::=	<i>Alle mulige søgekoder minus "år"</i>
ÅrPræfiks	::=	år = $\langle \text{ÅrIntervalListe} \rangle$
ÅrIntervalListe	::=	$\langle \text{ÅrInterval} \rangle \mid$ $\langle \text{ÅrIntervalListe} \rangle \langle \text{Seperator} \rangle \langle \text{ÅrInterval} \rangle$
ÅrInterval	::=	$\langle \text{År} \rangle \mid \langle \text{År} \rangle - \langle \text{År} \rangle$
År	::=	$\langle \text{Heltal} \rangle \langle \text{Heltal} \rangle \langle \text{Heltal} \rangle \langle \text{Heltal} \rangle$
TermListe	::=	$\langle \text{Term} \rangle \mid$ $\langle \text{TermListe} \rangle \langle \text{Seperator} \rangle \langle \text{Term} \rangle$
Term	::=	" $\langle \text{Streng} \rangle$ " \mid $\langle \text{BogstavListe} \rangle$
Streng	::=	$\langle \text{SBogstav} \rangle \mid \langle \text{Streng} \rangle \langle \text{SBogstav} \rangle$
BogstavListe	::=	$\langle \text{SBogstav} \rangle \mid \langle \text{BogstavListe} \rangle \langle \text{SBogstav} \rangle$
Bogstav	::=	<i>Alle tegn undtagen mellemrum, =, (,) og ", "</i>
SBogstav	::=	<i>Alle tegn undtagen "</i>
Seperator	::=	" " \mid " , "
Heltal	::=	[0-9]

Figur 4.22: BNF der definerer søgesystemets forespørgselssprog.

En forespørgsel kan bestå af et eller flere elementer. Hvert element kan bestå af en eller flere præfikser efterfulgt af en eller flere termer, eller af en eller flere termer uden præfiks. Hvis en term i en forespørgsel ikke eksplicit har et præfiks tilknyttet betyder det, at termen blot skal findes et eller andet sted i objektet. For eksempel vil forespørgslen "em,ti = (A B C)" betyde, at der skal søges efter objekter, hvor termerne "A samt B samt C" findes i emneord eller titel. Forespørgslen "A B C" betyder, at der skal findes objekter, hvori "A samt B samt C" findes. Parentesen i det første eksempel "em,ti = (A B C)" benyttes til at gruppere "A B C". Uden parentesen ville forespørgslen "em,ti = A B C" betyde at der skulle søges efter objekter, der har A i emneord eller titel samt B samt C.

Kapitel 5

Implementering af prototype

I dette kapitel gennemgås implementeringen af prototypen. Jeg har udvalgt en række forskellige eksempler, som jeg mener er vigtige, idet jeg ikke mener, at en fuldstændig gennemgang er relevant. Prototypen kan afprøves på adressen <http://www.isl.ruc.dk/cgi-bin/giss>.

5.1 Database

I forrige kapitel designede jeg en datamodel til repræsentation af data, indksering og viden i søgesystemet. Figur 4.11 på side 37 er et E/R-diagram, der viser denne model. Transformation fra E/R-diagrammet til databasen benytter en ofte anvendt metode, hvor entiteter og mange-til-mange relationer bliver til selvstændige databasetabeller, mens en-til-mange relationerne kan defineres som attributter. Herunder findes de databasetabeller, E/R-diagrammet i 4.11 på side 37 vil give med denne transformationsmetode:

```
SØGEKODE(SØGEKODE, BESKRIVELSE)
BASEFORMAT(BFID, NAVN)
BASE(BFID, BID, NAVN)
DELFELT(FELT, DELFELT, BESKRIVELSE)
BASEDEF(BFID, FELT, DELFELT)
SØGEDEF(FELT, DELFELT, SØGEKODE)
VISFORMAT(VID, NAVN, FORMATDEF)
VISDEF(FELT, DELFELT, VID)
OBJEKT(OID, BID)
DATA(OID, FELT, DELFELT, DATA, LINE)
TERM(TID, TERM)
INDEKS(OID, TID, SØGEKODE)
ORDBOG(ORDID, ORD)
TERMORD(TID, ORDID)
BØJNING(ORDID, ORD, FORM)
```

```

RELATIONSTYPE(RID, BESKRIVELSE)
RELATION(TID1, TID2, RID, GRAD)
RELDEF(RID, SØGEKODE)

```

Tabellerne oprettes i databasen ved hjælp af Structured Query Language (SQL) [Lorentz, 2000]. I figur 5.1 vises SQL-koden, der opretter tabellen BASE med attributten *BID* som primærnøgle og danner den relationelle afhængighed mellem BASE og BASEFORMAT ved at definere, at attributten *BFID* i BASE er en fremmednøgle, der relaterer til BASEFORMAT's *BFID*. Ved at definere disse afhængigheder i databasen, kan databasen kontrollere, at de overholdes. For eksempel er det ikke muligt, efter oprettelsen af fremmednøglen i BASE, at indsætte data i BASE, der ikke relaterer til et BASEFORMAT.

```

CREATE TABLE BASE(
  BID    INTEGER NOT NULL,
  NAVN   VARCHAR2 (30),
  BFID   INTEGER,
  CONSTRAINT BASE_PK PRIMARY KEY(BID));

ALTER TABLE BASE ADD CONSTRAINT BASE_BASEFORMAT_FK
FOREIGN KEY(BFID) REFERENCES GISS.BASEFORMAT(BFID);

```

Figur 5.1: Et eksempel på SQL-kode, der opretter tabellen BASE og derefter opretter afhængigheden mellem BASE og BASEFORMAT.

5.2 Databehandling

En bibliografisk enhed, der tilknyttes prototypen, skal konverteres fra danMARC2-formatet til repræsentationen i databasen. Man kan vælge forskellige strategier for programmeringen af denne konvertering, hvor de primære overvejelser handler om, hvor meget der skal håndteres i databasen og hvor meget der skal løses udenfor. Jeg valgte en strategi, hvor så meget som muligt blev håndteret i et program udenfor databasen. Dette valg er truffet, fordi programmer kodet i C/C++ er langt hurtigere, når det drejer sig om traditionel programmering, end samme funktionalitet etableret internt i databasen.

Behandlingen af objekter foretages i et C/C++ program, der indlæser danMARC2-formatet fra en almindelig tekstfil. Objekterne behandles et af gangen, det vil sige at alle linier i danMARC2-formatet indlæses, objektet behandles, og derefter indlæses næste objekt. Samtidigt med at linierne i

danMARC2-formatet indlæses, indsættes de i en intern datastruktur, der giver mulighed for at søge efter indholdet i forhold til felter/delfelter.

Når objektet er behandlet, oprettes det i databasen, det vil sige der skal oprettes et objekt i

OBJEKT(OID, BID)

og objektets informationer skal indsættes i

DATA(OID, FELT, DELFELT, DATA, LINE)

Denne del af oprettelsen er forholdsvis simpel, for eksempel vil den første linie

001 00*a0747561*b820050*fa

fra objektet med OID = 2370017 blive til de rækker, der vises i table 5.1.

OID	FELT	DELFELT	DATA	LINIE
2370017	001	*a	0747561	1
2370017	001	*b	820050	1
2370017	001	*f	a	1

Tabel 5.1: Eksempel på indsættelse af data i tabellen DATA.

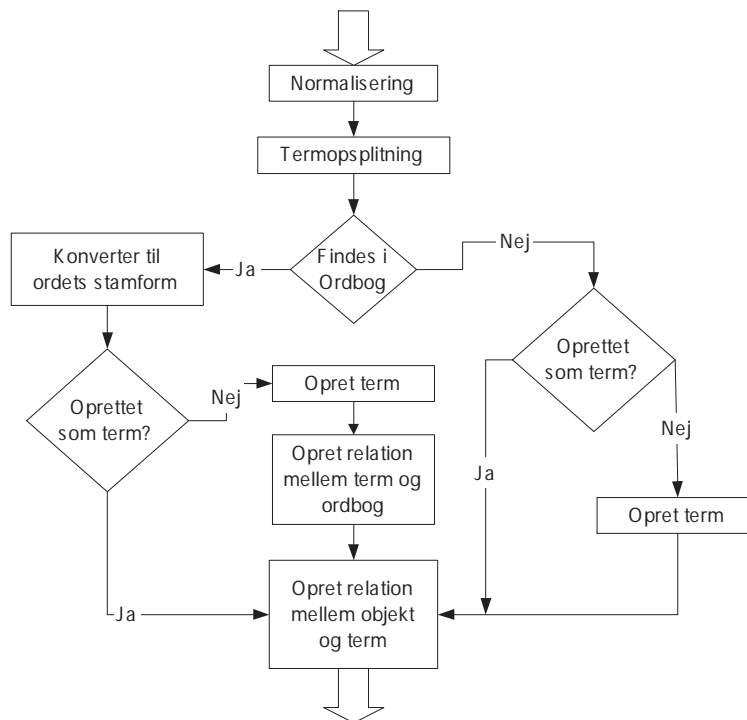
Herefter dannes beskrivelser af objektet i indekseringsprocessen, som skal indsættes i

INDEKS(OID, TID, SØGEKODE)

Indekseringsprocessen vises som flow-diagram i figur 5.2. *Normalisering* konverterer små bogstaver og substituerer specialtegn. For eksempel vil titlen ”Sm@å-Folks Helte & Heltinder” blive normaliseret til ”småfolks smaafolks helte og heltinder”¹. *Termopsplitning* opsplitter data i termer, hvor en term er kombinationer af tegn adskilt af mellemrum. For ovenstående eksempel vil det blive til følgende liste af termer {”småfolks”, ”smaafolks”, ”helte”, ”og”, ”heltinder”}.

Herefter behandles alle termene i den opsplittede liste. *Findes i ordbog?* undersøger, om termen er i ordbogen. Den del af termene, der findes i ordbogen, er specielle, fordi de udover at være termer også kan klassificeres som ord, og dermed vil kunne konverteres til stamform via ordbogen i *Konverter*

¹Tegnkombinationen @å definerer det gamle danske dobbelt a, *Smaa-Folks*. @å bliver til to termer i indekseringen *Små-Folks* og *Smaa-Folks*, således at objektet findes, uanset hvilket ”å” der indtastes.



Figur 5.2: Flow-diagram over indekseringsprocessen.

til ordets stamform. Ovenstående liste af termer {"småfolks", "smaafolks", "helte", "og", "heltinder"} bliver konverteret til {"småfolk", "smaafolks", "helt", "og", "heltinde"}². Uanset om termen findes i ordbogen eller ej, er næste del at afgøre, om termen allerede findes i systemet. Dette gøres i *Oprettet som term?*. Hvis termen ikke findes i systemet, oprettes termen i *Opret term*. Hvis termen er en del af ordbogen, oprettes en relation mellem identifikationen af ord i ordbogen og identifikation af termer i *Opret relation mellem term og ordbog*. Til sidst dannes en relation mellem termen og objektet i *Opret relation mellem objekt og term*, hvis der ikke allerede findes en sådan relation for den givne felt-/delfeltkombination.

Denne proces er kostbar, fordi det er nødvendigt at foretage opslag i databasen for at afgøre, om termen findes i ordbogen, og om den eksisterer i systemet. Efterfølgende indsætte data, skal disse opslag være hurtige, er det nødvendigt med en databaseindeksning på databasetabellerne. Det betyder, at opslaget bliver meget hurtigt, men til gengæld skal indekseringen vedligeholdes, når der tilføjes nye data til systemet. Dernæst er det nødvendigt at undersøge, om termen allerede er tilknyttet objektet, så der ikke bliver

²Termen *småfolks* findes i ordbogen og kan derfor konverteres til stamform, mens termen *smaafolks* forbliver uændret, fordi den ikke findes i ordbogen.

oprettet flere relationer fra en term til samme objekt. Uanset optimering har det ikke været muligt at få den gennemsnitlige behandlingstid for et objekt ned under 1 sekund. Det er naturligvis hurtigt, når objekterne skal indlæses, parses og i gennemsnit har 34 forskellige termer tilknyttet, men når der er 10 millioner objekter, svarer det til en samlet behandlingstid på cirka 4 måneder.

Denne model vil sandsynligvis være hurtig nok til den daglige opdatering, da der kan behandles cirka 100.000 objekter om dagen, men i etableringsfasen er den for langsom. Jeg valgte derfor en anden metode, der i stedet for at indlæse data direkte i databasen transformerer data til tekstfiler. Data til DATA-tabellen i en tekstfil har formatet

OID, FELT, DELFELT, DATA, LINIE

og data til INDEKS-tabellen i en tekstfil med formatet

OID, SØGEKODE, ORDID, TERM

Disse to filer kan trækkes ind i databasen ved hjælp af værktøjet SQL*Loader [Rich, 1999]. DATA-tabellen kan oprettes direkte fra informationerne i tekstfilen, mens informationerne til INDEKS-tabellen skal trækkes ind i en temporær tabel. Fra denne temporære tabel kan TERM-tabellen udledes som alle de unikke termer og indsættes i

TERM(TID, TERM)

dernæst kan der dannes relationer mellem termer og ordbog

TERMORD(TID, ORDID)

og afslutningsvis kan INDEKS-tabellen oprettes. På denne måde kan etableringen af objekter og indeksering i prototypen etableres på cirka en uge.

5.3 Forespørgselsevaluering

Behandlingen af forespørgsler starter med en CCL-parsing. CCL-parseren er en top-down parser, der benytter definitionen af prototypens forespørgselsprog i tabel 4.22 på side 50 som grundlag for at parse forespørgslerne³. Dette vil resultere i en liste af argumenter. For eksempel vil forespørgslen ”find objekter der i titel eller emneord har ungdom og metadon tilknyttet”:

³Præfikset *år* og behandlingen af årstalsintervaller er ikke implementeret.

Efter CCL-parsingen benyttes, for de enkelte argumenter, samme normaliseringsregler og opsplitning i termer som i behandlingen af objekter. I figur 5.3 vises et flow-diagram over forespørgselsevalueringen, hvor de dele, der er sammenfaldende med behandlingen af objekter, er markeret.

Efter *Normalisering* og *Termopsplitning* undersøges det først, om termen findes i *Oprettet som term?* Hvis dette er tilfældet, kan termen benyttes i den videre evaluering. Findes termen ikke, undersøges det, om den *Findes i ordbog?* Bliver termen fundet i ordbogen, konverteres den til stamform i *Konverter til ordets stamform*, og det undersøges, om denne term findes i *Oprettet som term?* Hvis den bliver fundet gennem denne konvertering via ordbogen, benyttes den i den videre evaluering. Hvis termen ikke findes i systemet eller via ordbogen ignoreres den, fordi der så ikke findes nogle objekter med den givne term.

Den næste del af forespørgselsevalueringen afhænger af brugernes valg. Hvis *Brug fuzzy-evaluering?* ikke er valgt, vil argumenterne blive fortolket med logisk *og*. Det vil sige at eksemplet fra før bliver til

$$\{(em, ti : ungdom) \text{ og } (em, ti : metadon)\}$$

og objekter, der opfylder forespørgslen, findes i *Find objekter* og *Aggregering*.

Hvis brugerne har valgt *Brug fuzzy-evaluering?* vil forespørgslen blive ekspanderet i *Find ekspanderende termer*, hvis dette er valgt. I ekspandering af forespørgslen udvides den med de termer, der relaterer til de enkelte argumenter. I eksemplet fra før kan første argument $(em, ti : ungdom)$ for eksempel ekspanderes til

$$em, ti : \{ \\ (1.0000, ungdom), \\ (0.1986, ung), \\ (0.1133, barn) \\ \}$$

og andet argument $(em, ti : metadon)$ til

$$em, ti : \{ \\ (1.0000, metadon), \\ (0.3807, narkomani), \\ (0.3092, narkomanforsorg), \\ (0.2497, stofmisbruger), \\ (0.1903, narkoman), \\ (0.1419, behandling), \\ (0.1186, narkotika), \\ \}$$

(0.1070, stofmisbrug)
}

For at kunne foretage aggregeringen skal hvert argument evalueres selvstændigt, det vil sig at alle objekter, der opfylder det første argument, findes og dernæst dem, der opfylder det andet argument, et cetera. Et objekt kan, hvis der er ekspanderet, for et enkelt argument matche mere end en term, hvilket betyder at objektet kan opfylde argumentet med forskellige grader. I eksemplet fra før, hvor *ungdom* blev ekspanderet med *ung* og *barn*, kan et objekt for eksempel opfylde både *ung* og *barn* og dermed opfylde argumentet til graden 0.1986 og 0.1133. Det er den højeste grad, hvormed objektet opfylder argumentet, der benyttes, således at et objekt kun kan opfylde et argument en gang.

Objekt	Grad	Argument
O_1	1	1
O_1	1	2
O_2	1	1

Tabel 5.2: Eksempel på resultatet, der skal benyttes til aggregering.

Sorteres denne liste efter objekter og hvilke argumenter de opfylder, vil aggregeringen kunne foretages ved at gennemløbe listen og beregne, i hvor høj grad objekterne opfylder forespørgslen. I tabel 5.2 vises et eksempel hvor de første to rækker er samme objekt der opfylder hver sit argument. Hvis forespørgslen for eksempel har to argumenter, vil objektet O_1 opfylde begge argumenter, mens O_2 kun opfylder det første argument.

Dette betyder, at aggregeringen for objekterne i tabel 5.2 vil kunne udregnes med gennemsnitsaggregeringen på følgende måde

$$\begin{aligned} owa_{\alpha=(\frac{1}{2}, \frac{1}{2})}(F, O_1) &= (\overbrace{1/2 * 1}^{\text{argument}_1} + \overbrace{1/2 * 1}^{\text{argument}_2}) = 1 \\ owa_{\alpha=(\frac{1}{2}, \frac{1}{2})}(F, O_2) &= (1/2 * 1 + 1/2 * 0) = 1/2 \end{aligned} \quad (5.1)$$

hvor argumenterne er vist, og det kan ses, at O_1 matcher begge argumenter, mens O_2 kun matcher det første. Aggregeringen resulterer, i en liste af objekter med tilhørende grad af opfyldelse, der kan sorteres faldende efter denne grad. I eksemplet (5.1) ville listen være $\{(1.000, O_1), (0.500, O_2)\}$, og sorteringen ville i dette eksempel ikke ændre på rækkefølgen.

Kapitel 6

Diskussion

Det overordnede mål med dette projekt har været at undersøge, om et fleksibelt intelligent søgesystem kan understøtte bibliotekarers ønsker og behov i forbindelse med bibliografisk søgning. Herunder at undersøge, hvordan inddragelse af kommunikative metoder kan bidrage med en indsigt i målgruppen, der kan benyttes i forbindelse med udvikling af en prototype. Projektet skulle beskrive den første fase i udvikling af et søgesystem fra den første kontakt til målgruppen frem til en afprøvning af første prototype.

I dette kapitel diskuteres projektets forskellige dele. Kapitlet begynder med at beskrive og diskutere afprøvningen af prototypen, herefter diskuteres forskellige dele fra kapitlerne Design af prototype, Implementation og Interviewundersøgelse. Afslutningsvis konkluderes projektet.

6.1 Afprøvning

Målet med den afsluttende afprøvning af prototypen var at vurdere denne som helhed. Men i interviewundersøgelsen forud for konstruktion af prototypen var den generelle holdning til forespørgselsekspansion forholdsvist afvisende, hvilket kunne pege i retning af, at målgruppen ikke ønskede eller havde behov for sådanne funktionaliteter. Jeg valgte derfor at ændre målsætning for afprøvningen, således at målet blev en specifik vurdering af associationsnet og ekspansion af forespørgsler for at undersøge målgruppens holdning, når vurderingsgrundlaget blev mere konkret.

Af ressourcemæssige årsager valgte jeg to informanter ud til afprøvning, en fra Roskilde bibliotek (RB) og en fra Roskilde Universitetsbibliotek (RUb). Informanterne fra Dansk BiblioteksCenter (DBC) blev fravalgt, fordi begge informanter allerede kendte til dele af teknikken fra forskningsprojektet ”*Fleksibel søgning i DanBib*” og derfor måske var forudindtaget. De to udvalgte informanter blev udvalgt ud fra deres holdninger til forespørgselseks-

pansion i interviewundersøgelsen. Den ene havde en meget klar holdning om, at ekspansion under ingen omstændigheder ville kunne bruges i de søgninger, bibliotekarer udfører. Den anden var lidt mere moderat og kunne godt få øje på situationer, hvor det måske kunne være en hjælp.

Afprøvningen blev overordnet tilrettelagt således, at jeg først introducerede prototypens funktionalitet og begrænsninger og viste nogle eksempler på, hvad prototypen kunne, og hvilken indflydelse det har på forespørgselsevurderingen. Herefter prøvede informanterne selv at udføre søgninger. Til sidst blev informanterne bedt om at vurdere forespørgselsekspansion på ny.

6.1.1 Eksempler

Jeg valgte at introducere associationsnettet først, fordi det er grundlaget for ekspansionen af forespørgslerne og derfor en nødvendig forudsætning for at forstå denne. Jeg brugte associationsnavigatoren¹ til introduktion af associationsnettet. På denne måde kunne informanterne selv prøve at navigere rundt i relationerne mellem termer, og dermed få en fornemmelse af, hvordan nettet er opbygget.

For at eksemplificere forespørgselsekspansionen skulle der udvælges forespørgsler, hvor inddragelse af relationerne i associationsnettet var tydelige. Jeg valgte to eksempler, forespørgslerne ”*ungdom metadon*” og ”*ungdom narkomani*”. Den første forespørgsel, ”*ungdom metadon*”, vil uden ekspansion ikke finde nogle objekter. Med fuld ekspansion får de første 32 objekter en relevans, der er større end blot, at en af termerne findes i objektet.

1. GRAD: 0.690327 ObjektID: 6517392	2. GRAD: 0.690327 ObjektID: 8974717
001 00 *a73230101	001 00 *a04265130
008 00 *a1988 *ldan	008 00 *a1965 *ldan
009 00 *gpxx	009 00 *gpxx
032 00 *aDAR198908 *aABU198937 *aDAR *aABU	032 00 *xSFD197075 *xSFD
100 00 *aKoch *hIda	245 00 *aBuket for bilister *cbundet af Erik Seidenfaden
245 00 *a80'ernes ungdom og speed	260 00 *bGyldendal *c1965
557 00 *aUnge pædagoger *v1988, nr. 8 *z0106-5386	505 00 *aAntologi
630 00 *fSpeed *funge *unarkomani *fnarkomani	530 00 *aIndhold: Ilja Ehrenburg: Paris år 1900. Johannes V. Jensen: Den barmhjertige samant. Lee Strout White: Farvel til min ungdoms elskede. Jens Kruuse: Man klarer sig.
652 00 *m61.648	Aldous Huxley: Rejsen. Robert Storm Petersen: Færdsel. Erik Seidenfaden: Det firljulede menneske. Halfdan Rasmussen: Gi' den en tand til! Pierre Daninos: Mine kørselsuheld. Finn Gerdes: Hval i en åluse. Jesper Ewald: Hertuginde af Delage. Birgitta Stenberg: Raggartiv. Lise Nørgaard: Campingturen. Knud Sønderby: Fartens narkomani
m01 00 *ate	652 00 *msk *p88.02
	700 00 *aSeidenfaden *hErik
	m01 00 *ano

Figur 6.1: De første to objekter i resultatet af forespørgslen ”*ungdom metadon*”.

I figur 6.1 vises de to første objekter i svaret, hvor det første har titlen ”*80'ernes ungdom og speed*” og handler om unge og narkomani. Det andet objekt har titlen ”*Buket for bilister*” og handler om fartens narkomani.

¹Se afsnit 4.3.4 på side 41.

Det første objekt virker som en fornuftig erstatning for *ungdom* og *metadon*, mens det andet er lidt misvisende, men ikke egentlig støj, fordi det er muligt at se sammenhængen mellem *metadon* og *narkomani*. Objektet optræder i svaret, fordi termen *narkomani* benyttes som metafor, hvilket er grunden til at objektet er lidt misvisende. Den model, prototypen benytter til beskrivelse af objekterne, er ikke avanceret nok til at kunne behandle metaforer. Dette ville kræve avanceret natursprogsanalyse og dermed en anden beskrivelsesmodel. De næste to objekter vises i figur 6.2. Selvom det andet objekt er lidt misvisende, er dette fornuftige alternative svar på forespørgslen.

3. GRAD: 0.654613 ObjektID: 7369727	4. GRAD: 0.624858 ObjektID: 7533511
001 00 *a71160963	001 00 *a72561767
008 00 *a1984 *Idan	008 00 *a1987 *Idan
009 00 *gxx	009 00 *gxx
032 00 *aDAR198403 *aDAR	032 00 *aDAR198700 *aDAR
100 00 *aBache *hJørgen	100 00 *aKristiansen *hTom
245 00 *aGodt med i Ringkøbing Amt *eJørgen Bache, interview med Bernhard Lodahl	245 00 *aSådan kan idrætten hjælpe stofmisbrugere
504 00 *aRingkøbing Amts ungdoms- og misbrugsafdeling	557 00 *aDansk ungdom og idræt *vÅrg. 90, nr. 5 (1987) *z0045-9631
557 00 *aAlkohol-debat *v1984, nr. 15 *z0106-3033	630 00 *fidrætsterapi *fstofmisbrug *fjeldvandring *upsiatri
630 00 *falkoholikerforsorg *fmarkomanforsorg	652 00 *m61.648
633 00 *aRingkøbing Amt *u social forsorg	
652 00 *m38.8	
700 00 *aLodahl *hBernhard	
m01 00 *ate	

Figur 6.2: Objekt 3 og 4 i svaret på forespørgslen ”*ungdom metadon*”.

I det andet eksempel, ”*ungdom narkomani*”, finder prototypen to objekter uden inddragelse af ekspansion, netop de to første objekter fra den første forespørgsel, som vises i figur 6.1. Ekspansion af denne forespørgsel tilføjer objekter, som opfylder forespørgslen til en vis grad. For eksempel er titlen i det tredje objekt, som vises i 6.3, *Unge forbrug af rusmidler i Nykøbing F.*, hvilket også er et fornuftigt alternativt svar på forespørgslen.

3. GRAD: 0.599282 ObjektID: 735871
001 00 *a8798399403
008 00 *a1991 *Idan
009 00 *gxx
021 00 *a27-983994-0-3
245 00 *aUnge forbrug af rusmidler i Nykøbing F.
260 00 *a[Nykøbing F.] *h[SSP-gruppen i Nykøbing F.] *c[1991]
*grekvirering ... SSP-sekretariatet
512 00 *aRapporten er udarbejdet af en arbejdsgruppe under SSP-gruppen
652 00 *p66.88 *p66.83 *p30.1664
661 00 *aalkoholisme *aDanmark *anarkomani *arygning *astatistiske data *aunge
710 00 *aNykøbing Falster *cSSP-gruppen
910 00 *aSSP-Gruppen i Nykøbing F.
m01 00 *ate

Figur 6.3: 3. objekt i svaret på forespørgslen ”*ungdom narkomani*”. (De to første objekter vises i figur 6.1)

Dette eksempel viser også, at forskellen på evaluering med boolesk *og* ekspansion er, at der ved ekspansion kan tilføjes objekter, der opfylder forespørgslen til en vis grad. Objekter, der findes ved evaluering med boolesk *og* vil altid optræde som de første objekter i svaret ved en ekspanderet evaluering, fordi de netop er de eneste objekter, der kan opfylde forespørgslen

til graden 1².

For at vise effekten af inddragelse af en ordbog i søgesystemet, valgte jeg forespørgslen ”*barn salmebog*”. Det første objekt i svaret vises i figur 6.4, hvor det kan ses, at titlen er *Børnenes salmebog*. Det viser, hvordan ordbogen benyttes til at konvertere mellem forskellige bøjningsformer, her en konvertering fra *barn* til *børnenes*. Informanterne er vant til at benytte tegnmaskering til at løse denne type problemer, men de var enige om, at det er vanskeligt, når ord skifter stamme ved bøjning, for eksempel fra *barn* til *børn*. Det er langt nemmere, når det alene er endelserne, der maskeres, fordi dette kan gøres ved en trunkering. For eksempel vil *løve?* matche alle termer, der begynder med *løve*.

```
1. ObjektID: 519
001 00 *a8711004991
008 00 *a1971 *Idan *z1992
009 00 *gxx
021 00 *a8711004991
245 00 *aBørnenes salmebog *cudvalg af salmer og bibelske sange
for skolen og hjemmet *esamlet og udgivet af K. Byrjalsen
260 00 *aKbh. *bAschehoug *c1973
700 00 *aByrjalsen *hK.
m01 00 *adi
```

Figur 6.4: Første objekt i svaret på forespørgslen ”*barn salmebog*”.

6.1.2 Informanternes afprøvning af prototypen

I den næste fase skulle informanterne selv afprøve prototypen. Det viste sig, at informanterne havde svært ved at fremkomme med forespørgsler, der gav svar, som belyste funktionaliteten. Dette kunne skyldes, at prototypen ikke understøtter samme funktionalitet, som de applikationer informanterne normalt anvender, eller fordi der i afprøvningsseancen ikke var tid til at blive fortrolige med prototypen. Informanterne var inden afprøvningen gjort bekendt med det overordnede indhold i afprøvningen, men havde naturligvis ikke mulighed for at forberede sig, hvilket kunne betyde at informanterne følte sig presset og derfor afsøgte problemstillinger, de kendte, men som krævede funktionalitet, der ikke var understøttet. For eksempel valgte en af informanterne som første forespørgsel at finde bøger af Karen Blixen, som ikke er skrevet under et af hendes pseudonymer. Dette ville kræve, at prototypen understøtter operatoren *ikke* og kendskab til Blixens pseudonymer, for eksempel forespørgslen ”*blixen og ikke (dinesen eller andrézel eller osceola)*”.

I indledningen³ argumenterede jeg for, at det var vigtigt at udvælge datagrundlaget med omhu, fordi erfaringer fra projektet ”*Fleksibel søgning i DanBib*” havde vist, at det var vanskeligt at benytte projektets prototype til det daglige arbejde, og derfor vanskeligt at etablere en god evaluering. Det er derfor nødvendigt at konstruere afprøvningen og opfinde opgaver, der skal løses af testgruppen, hvilket kræver en tålmodighed og en vilje, som ikke

²Se definitionen af funktion til beregning af associationsgrad i udtrykket (4.3) på side 44.

³Se afsnit 1.3 på side 7.

kan forventes at findes i en tilfældig udvalgt testgruppe. Det er min overbevisning, at den bedste evaluering af søgesystemer opnås, hvis informanterne kan anvende søgesystemet i deres daglige arbejde, fordi det system, der skal evalueres, derved kan erstatte de eksisterende i en periode. Det ville i hvert fald i denne afprøvning have betydet, at det havde været lettere for informanterne at afprøve prototypen.

1. GRAD: 1 ObjektID: 2238206	2. GRAD: 0.787446 ObjektID: 4864367
001 00 *aX644634498	001 00 *a0598544
008 00 *ldan	008 00 *a1980 *ldan
009 00 *g:xx	009 00 *g:xx
100 00 *aNielsen *hJørgen Chr.	100 00 *aHANSEN *hJOHN
245 00 *aEn gennemgang af den spanske anarkistbevægelses strukturelle og ideologiske udvikling, fra Bakunin frem til borgerkrigen	245 00 *aPOUM OG REVOLUTIONEN UNDER DEN SPANSKE BORGERKRIG - ENHEDSARBEJDE, REGERINGSDELTAGELSE OG SPØRGSMÅLET OM DEN POLITISKE MAGT.
kompletivsætninger efter overordnet verbal i fortid	260 00 *aKbh *c1980
558 00 *aSpanske specialer *x8234326	559 00 *aSpeciale til kandidateksamen *aBedømmelsesdato: 000080
630 00 *adansksproget *apecialer,	630 00 *hHISTORIE
spansk *aanarkisme *aSpanien *apolitiske historie	652 00 *m97.4 *m32.16 *m32.08 *m32.2 *m32.24
m01 00 *ate	631 00 *hHISTORIE
	652 00 *hSPANIENS HISTORIE *hMARXISME, SOCIALISME OG KOMMUNISME *hPOLITISK TEKNIK OG PROPAGANDA *hPOLITISKE SYSTEMER *hPOLITISKE PARTIER
	m01 00 *aex

Figur 6.5: Det to første objekter i svaret på forespørgslen ”spanske borgerkrig speciale anarkisme”.

Selvom der var visse vanskeligheder i denne fase, lykkedes det dog for informanterne at fremkomme med forespørgsler, der viste prototypens kvaliteter. En af grundene hertil var, at prototypen ikke havde problemer med, hvor mange argumenter der blev indtastet, fordi den evaluerer forespørgslen ved at finde de objekter, der opfylder den bedst. De systemer informanterne normalt anvender vil ikke fremkomme med nogle objekter, hvis blot et af argumenterne ikke findes⁴. Et eksempel var en forespørgsel efter specialer, der omhandler den spanske borgerkrig og anarkisme, ”spanske borgerkrig speciale anarkisme”. Uden ekspansion fandt prototypen et objekt, der opfyldte denne forespørgsel. Forespørgselsekspansionen tilføjer objekter gennem en relation mellem *anarkisme* og *marxisme*. I figur 6.5 vises de første to objekter i svaret, hvor det andet objekt er resultat af ekspansionen. Denne tilføjelse fandt informanten både relevant og brugbar, primært fordi det giver en ide om, hvordan man kan komme videre, hvis det ene objekt, der opfyldte forespørgslen uden ekspansion, ikke var tilfredsstillende.

6.1.3 Opsummering

Til sidst i afprøvningen vendte jeg tilbage til informanternes holdning til forespørgselsekspansion, efter at de havde set og prøvet det i praksis. Det var klart og utvetydigt, at afprøvningen havde skabt en holdningsændring

⁴Hvis der anvendes boolesk *og* mellem argumenterne, hvad der ville være normalt for sådanne forespørgsler.

hos informanterne. De havde nu ikke de store forbehold og mente begge, at ekspansion af forespørgsler ville være et værktøj, de ville benytte. De to klarreste elementer i deres positive tilbagemelding var muligheden for at kunne fravælge og styre ekspansionen samt kvaliteten af de udvidelser, prototypen fremkom med. Herudover var de begge imponeret af den effekt, inddragelse af ordbogen gav, og kunne i denne forbindelse se perspektiver for den almindelige bruger, der ofte ikke ved, at emneord i bibliografiske søgesystemer altid optræder i ubestemt flertal, altså *biler* og ikke *bil*.

Afprøvningen viste, at der er grundlag for at arbejde videre med prototypens funktionaliteter i forhold til målgruppen. Den viste også en række problemer, der bør diskuteres og vurderes i den videre udvikling af prototypen. I forbindelse med afprøvning er det vigtigt, at fokus ligger på det, man gerne vil afprøve, og ikke bliver styret af for eksempel manglende funktionalitet. Det er derfor nødvendigt at designe prototypen således, at den kan opfylde informanternes vante funktionalitet, eller at afprøvningen tilrettelægges, så denne mangel ikke bliver forstyrrende.

De videre afprøvninger af prototypen vil i højere grad skulle klarlægge specifikke holdninger til bestemte dele af prototypen, og ikke som her, undersøge overordnede holdninger, hvilket derfor stiller andre krav til strategien omkring selve afprøvningen. Brugernes adfærd i forbindelse med brugen af grænsefladen kan med fordel undersøges sammen med informanterne, mens søgesystemets funktionalitet bør afprøves over længere tidsperioder, hvor prototypen indgår i informanternes daglige arbejde.

6.2 Design og implementation

Et af delmålene med prototypen var at skabe et søgesystem, der kunne behandle objekter med forskellig strukturel formatering og generisk tilknytning af viden. Ved at modificere danMARC2-formatet, så det kan repræsentere fuldtekstobjekter, er den første del af målsætningen opnået. Samtidig har indførelse af metainformation skabt en repræsentation, der er fleksibel og rummer mulighed for repræsentation af forskellige formater med samme struktur som danMARC2-formatet.

Den generiske tilknytning af viden er opnået gennem en repræsentation, der kan håndtere vægtede relationer. Dette giver mulighed for at tilknytte forskelligartet viden og for eksempel lade brugeren afgøre, hvilken der skal inddrages ved evaluering af forespørgsler. Modellen understøtter mulighed for at kunne beskrive graden af sammenhæng mellem termerne ved hjælp af vægte. En anden mulighed, der måske skulle overvejes i den videre udvikling, er mulighed for at kunne vægte de forskellige typer af viden i forhold

til hinanden. Et eksempel kan være at vægte synonymrelationer højere end relationer i associationsnettet.

I forskningsprojekter er det ofte interessant at afprøve teorier, metoder og teknikker i forskellige sammenhænge. For forskning i søgesystemer vil dette ofte være at undersøge, om modellen kan håndtere andre datagrundlag. Den fleksibilitet, der er skabt i prototypen, gør den velegnet til dette formål, fordi dette kan gøres uden at skulle ændre på prototypen. Den generisk tilknyttede viden vil på samme måde kunne benyttes til at afprøve forskelligartet viden på samme eller forskellige datagrundlag.

Associationsnettet i denne prototype er dannet på baggrund af en udvalgt mængde af indholdet i objekterne. Det er efter etableringen blevet klart, at de valg, jeg havde truffet, ikke helt har elimineret støj, for eksempel har det været et problem, at formateringen af forfatternavne har været forkert, og istedet for at tilknytte *Jensen*, *Jens* er det kun *Jensen*, der er med. Dette er en klar fejl, men meget tidskrævende at ændre, fordi det vil kræve, at hele repræsentationen etableres påny.

Den centrale del i forespørgselsevalueringen er OWA-aggregeringen, der skaber muligheden for den fleksible evaluering af sammensatte ekspanderede udtryk. I denne prototype fastlåses ordensvægtene, således at der benyttes en gennemsnitsfunktion svarende til en evaluering, der ligger mellem boolesk *og* og *eller*. Dette kan uden videre generaliseres således, at evalueringen kan kontrolleres dynamisk og dermed give brugerne mulighed for at bestemme, hvordan systemet skal evaluere forespørgslerne. Det er dog mere relevant at forstille sig, at der overordnet for hele søgesystemet kan justeres på evalueringfunktionen, fordi det er uhyre vanskeligt at formidle denne mulighed til brugerne. I projektet "*Fleksibel søgning i DanBib*" forsøgte vi på et tidspunkt at tilbyde denne funktionalitet i brugergrænsefladen ved en "*skyder*", der kunne skydes fra *og* til *eller*. Evalueringen kunne derved justeres i et "interval" fra *og* til *eller*. Flyttes "*skyderen*" fra *og*, mod *eller*, ville evalueringen inddrage flere objekter i svaret, og mod *og*, færre. Der var delte meninger om denne funktionalitet. Nogle kunne anvende og forstå konceptet umiddelbart, mens andre havde svært ved at forstå mulighederne i den.

OWA-aggregering vil betyde, at evalueringen bruger mere tid end ved den konventionelle booleske søgning. Det vil derfor være oplagt at undersøge, hvilke optimeringsmuligheder der er. I prototypen viste det sig, at det var hurtigere at trække informationerne ud af databasen og foretage aggregeringen i et C/C++ program, end hvis databasen skulle foretage aggregeringen. Det kan umiddelbart virke underligt, fordi databasen jo netop burde kunne optimere forespørgslerne, og måden hvorpå disse evalueres. Men databasen er konstrueret, så den kan håndtere dynamiske forespørgsler af for-

skellig karakter, og vil derfor benytte generelle løsningsmodeller. Et specifikt C/C⁺⁺ program, der håndterer aggregeringen, skal ikke kunne andet, fordi typen af forespørgsler er kendt på forhånd. Det betyder, at fortolkning af forespørgsel samt evalueringsstrategi ikke skal håndteres på forespørgselstidspunktet, men kan indlejres i programmeringskode.

I eksperimentelt arbejde med prototyper bliver metoder og modeller evalueret og ændret løbende, og det kan derfor være problematisk med meget store datagrundlag. I denne prototype var behandlingstiden for etablering af databasen cirka en uge. Det gør processen mindre fleksibel og stiller krav til afprøvning på mindre delmængder af datagrundlaget, fordi det er tidskrævende at ændre på fejl i det fulde datagrundlag. Det er dog min overbevisning, at datagrundlaget på samme måde som funktionalitet er afgørende for, hvilke muligheder der er for afprøvning.

6.3 Interviewundersøgelse

Målet med interviewundersøgelsen var at få en indsigt i brugernes ønsker og behov i forbindelse med bibliografisk søgning, hvilket jeg mener er lykkedes. Man kan diskutere nogle af de valg, jeg traf i forbindelse med denne undersøgelse. Først og fremmest er jeg ikke en trænet interviewer og burde derfor have øvet mig lidt inden undersøgelsen. Eventuelt ved nogle pilot-interview, der kunne bruges til at få en større indsigt i målgruppen. De manglende kvalifikationer som interviewer træder også tydeligere igennem i kvalitative interviews med meget åbne spørgsmål, fordi der ikke er den samme køreplan som ved spørgeskemaundersøgelser. Det er usandsynlig vigtigt at være bevidst om sin rolle og for eksempel ikke lade sig rive med, fordi indholdet bliver spændende. Endelig måtte jeg erfare, at teknikken skal tjekkes mange gange, inden et interview foretages. Jeg havde en meget dårlig båndoptager, der skulle placeres præcist for at få både informant og interviewer med. Faktisk skulle man være opmærksom på, hvilket køn informanten havde. Hvis det var en kvindelig informant skulle mikrofonen være tættere på hende end på mig, mens dette for det ene interview, jeg lavede med en mandlig informant, betød at jeg var svær at aflytte, fordi mikrofonen så var for tæt på informanten.

For at vurdere undersøgelsens udsigelseskraft må man først kigge på undersøgelsens gyldighed og pålidelighed. I afsnit 3.1.2 på side 17 redegjorde jeg for de overvejelser, der skulle sikre både gyldighed og pålidelighed. På trods af mine manglende kvalifikationer som interviewer og de tekniske problemer, mener jeg, at undersøgelsen opfyldte de krav og forventninger, jeg havde. Hvorvidt undersøgelsen er generaliserbar er dog vanskeligere at vurdere. De udvalgte informanter dækker et bredt udsnit af bibliotekarere, men

det er måske for få til en egentlig generalisering. Omvendt var interviewene kvalitative og meget åbne, og der var ikke nogen væsentlige modsætninger i informanternes holdninger, som kunne så tvivl om en generalisering. Jeg vil derfor mene, at de ønsker og krav, jeg har benyttet i projektet, er gældende for hele målgruppen. Med hensyn til afprøvningen bliver det langt mere tvivlsomt om det holdningsskifte, der var blandt informanterne, er generaliserbart. En måde, der kunne have givet en større sikkerhed, ville have været at inddrage informanter, der ikke havde været med i interviewundersøgelsen, for at undersøge, om de havde samme holdning, som disse informanter kom frem til efter afprøvningen.

6.4 Konklusion

Uanset vanskelighederne med interviewundersøgelsen blev den viden, jeg fik om målgruppen vigtig, og den har haft indflydelse på en række valg i designprocessen. Der er nogle konkrete punkter, hvor det er muligt at se den direkte konsekvens af indsigten i målgruppen, for eksempel inddragelse af CCL, som jeg ikke havde valgt, hvis det ikke havde været et utvetydigt krav fra informanterne.

En anden indflydelse, som er mindst ligeså interessant, er den indirekte påvirkning i designprocessen. Der er en væsentlig forskel på at opfatte brugerne af søgesystemet som brugere i datalogisk forstand, der bruger et program, og det at opfatte dem som del af en kommunikationsproces, og dermed som modtagere. Den kommunikative for forståelse betød, at mit fokus ikke indsnævrede sig til den datalogiske bruger, men at forudsætningen for, at brugeren ønsker at benytte søgesystemet, er brugerens behov for information. Det betyder, at brugergrænsefladen ikke kun skal formidle systemets funktionalitet, den er også midlet til opfyldelse af en kommunikationsproces. Jeg er overbevist om, at en række af de valg, jeg har truffet, ville have været anderledes, hvis jeg ikke havde den kommunikative dimension med.

Indsigten i målgruppen har også stor indflydelse på afprøvningen, fordi den forståelse, der er opnået, betyder, at det billede, jeg havde af brugeren, i højere grad blev styret af ønsket om at forstå, hvorfor de gjorde, som de gjorde, og hvad de mente. Det blev derfor lettere at modtage kritikken af prototypen som noget konstruktivt, og derfor ikke ende med at skulle forsvare de trufne valg eller benytte udviklede tekniske forklaringer for at få ret. Udover en indsigt i målgruppen, der muliggør et design, der understøtter målgruppens ønsker og behov, har inddragelsen af kommunikative teorier og metoder også bidraget til selve udviklingsprocessen.

Svaret på det overordnede spørgsmål i problemformuleringen

Kan bibliotekarers behov, ønsker og krav til bibliografiske søgesystemer understøttes af et generisk intelligent søgesystem?

vil umiddelbart være et ja. Det er dog helt tydeligt, at bibliotekarerne skal være med som kompetente medspillere i udviklingsprocessen. Forløbet i projektet understreger også vigtigheden af de metoder, der anvendes til udvikling. Det holdningsskifte, der var blandt informanterne, efter de havde haft mulighed for at afprøve prototypen, understreger vigtigheden i brugen af konkrete og fungerende prototyper. Jeg er derfor overbevist om, at udvikling af søgesystemer med fordel kan benytte sig af prototyping, fordi det ofte er meget abstrakte problemstillinger, informanterne ellers skal forholde sig til.

Det søgesystem, der er designet i dette projekt, er meget fleksibelt og kan opfattes som en model, der kan benyttes på mange forskellige datagrundlag og med forskellig viden. Det vil derfor være muligt at undersøge, hvordan funktionaliteten virker i forskellige sammenhænge. Det er for mig indlysende, at den funktionalitet, der er anvendt, har et potentiale, der bør udforskes yderligere.

Litteratur

- [Andersen and Lindstrøm, 1997] Tim Frank Andersen and Martin Lindstrøm. *Mærkevarer på Internettet*. Børsens Forlag A/S, København, Danmark, 1997.
- [Andreasen and Bulskov, 1998] Troels Andreasen and Henrik Bulskov. *Fleksibel søgning i danbib - dokumentation*. Technical report, Roskilde Universitetscenter, 1998.
- [Andreasen *et al.*, 2000] T. Andreasen, J.F. Nilsson, and H.E. Thomsen. Flexible query answering systems - recent advances. In Henrik L. Larsen, Janusz Kacprzyk, Slawomir Zadrozny, Troels Andreasen, and Henning Christiansen, editors, *Proceedings of the Fourth International Conference on Flexible Query Answering Systems, FQAS' 2000, October 25-28, 2000, Warsaw, Poland*, LNAI, Heidelberg, October 22-28 2000. Physica-Verlag.
- [Andreasen, 1998] Troels Andreasen. Fuzzy logik i danbib. *Referencen*, 1, 1998.
- [Belden and Melnick, 1999] Eric Belden and Jack Melnick. *"ORACLE Call Interface - Programmer's Guide*. Oracle Corporation, United States, 1999.
- [Beyer and Holtzblatt, 1998] Hugh Beyer and Karen Holtzblatt. *Contextual Design*. Morgan Kaufmann Publishers, San Diego, USA, 1998.
- [Biblioteksstyrelsen, 1999] Biblioteksstyrelsen. *Praksisregler for søgeveje. Opsætning af søgeveje og udfoldning af koder under danMARC2*. Biblioteksstyrelsen, København, Danmark, 1999.
- [Brix *et al.*, 1998] Thomas Brix, Henrik Bulskov, and Jacob Ravn Nielsen. *Emailarkivering*. Technical report, Roskilde Universitetscenter, 1998.
- [Bulskov, 1998] Henrik Bulskov. *Informationssøgning - en sammenligning af søgeteknikker*. Technical report, Roskilde Universitetscenter, 1998.
- [Dansk standardiseringsråd, 1990] Dansk standardiseringsråd. *DS 2347, Information og dokumentation - Kommandoer til interaktiv tekstsøgning*. DS-tryk, Danmark, 1990.

- [DBC, 2001] DBC. Dbc's hjemmeside. <http://www.dbc.dk>, 2001.
- [Fiske, 1990] John Fiske. *Introduction to communication studies*. Routledge, NY 10001 New York, 2 edition, 1990.
- [Forsberg *et al.*, 1999] Allan Forsberg, Troels Andreasen, and Henrik Bulskov. En eksperimentel søgeflade til danbib. *VIDEN OM*, Særnummer, 1999.
- [Frøkjær, 1985] Erik Frøkjær. Systemudvikling via prototyper. <http://www.dat.ruc.dk/undervisning2/scripts/kursmain.php3?job=display&kurid=22>, 1985.
- [Josuttis, 1999] Nicolai M. Josuttis. *The C++ Standard Library - A Tutorial and reference*. Addison-Wesley, Reading, MA, USA, 1999.
- [Katalogdatarådet for Biblioteksstyrelsen, 1998a] Katalogdatarådet for Biblioteksstyrelsen. *danMARC2*. Dansk BiblioteksCenter, Ballerup, Danmark, 1998.
- [Katalogdatarådet for Biblioteksstyrelsen, 1998b] Katalogdatarådet for Biblioteksstyrelsen. *Katalogiseringsregler og bibliografisk standard for danske biblioteker*. Dansk BiblioteksCenter, Ballerup, Danmark, 1998.
- [Klir and Yuan, 1995] George J. Klir and Bo Yuan. *Fuzzy Sets and Fuzzy Logic - Theory and Applications*. Prentice Hall, United States, 1995.
- [Kvale, 1994] Steinar Kvale. *Interview - En intruduktion til det kvalitative forskningsinterview*. Hans Reitzels Forlag, Danmark, 1994.
- [Lorentz, 2000] Diana Lorentz. *ORACLE SQL Reference*. Oracle Corporation, United States, 2000.
- [Musciano and Kenney, 1998] Chuck Musciano and Bill Kenney. *HTML - The definitive Guide*. O'Reilly, United States, 1998.
- [Naumann and Jenkins, 1982] J. D. Naumann and A. M. Jenkins. *Prototyping: The New Paradigm for Systems Development*. MIS Quarterly, 1982.
- [Nielsen, 2000] Jacob Nielsen. *Designing Web Usability*. New Riders Publishing, United States, 2000.
- [Rich, 1999] Kathy Rich. *ORACLE 8i Utilities*. Oracle Corporation, United States, 1999.
- [Salton and McGill, 1983] Gerard Salton and Michael J. McGill. *Introduction to Mordern Information Retrieval*. McGraw-Hill, Inc., United States, 1983.

-
- [Salton, 1988] Gerard Salton. *Automatic Text Processing: The Transformational, Analysis, and Retrieval of Information by Computer*. Addison-Wesley, United States, 1988.
- [Sepstrup, 1999] Preben Sepstrup. *Tilrettelæggelse af information*. Forlaget Systime A/S, Danmark, 1999.
- [Østbye *et al.*, 1997] Helge Østbye, Knut Helland, Kari Knapskog, and Terje Hillesund. *Metodebok for mediefag*. Fagbokforlaget Vigmostad og Bjørke AS, Norway, 1997.
- [Stroustrup, 2000] Bjarne Stroustrup. *The C++ Programming Language*. Addison-Wesley, Reading, MA, USA, 2000.
- [Yager, 1988] R.R. Yager. On ordered weighted averaging aggregation operators in multicriteria decision making. *IEEE Transactions on Systems, Man and Cybernetics*, 18, 1988.