



KUNGL. TEKNISKA HÖGSKOLAN

Royal Institute of Technology
Numerical Analysis and Computing Science

TRITA-NA-D9807 • CID-34, KTH, Stockholm, Sweden 1998

Dokumentbaser som stöd för lärare

– en prototyp till en rikstäckande, distribuerad dokumentbas för skolbruk

Karl Hjalmarsson



CID
Centre for
User Oriented IT Design

Karl Hjalmarsson

Dokumentbaser som stöd för lärare

– en prototyp till en rikstäckande, distribuerad dokumentbas för skolbruk

Report number: TRITA-NA-D9807 , CID-34

ISSN number: ISSN 1403-0721

Publication date: September 1998

E-mail of author: d92-khj@nada.kth.se

URL of author: <http://www.student.nada.kth.se/~d92-khj/>

Reports can be ordered from:

CID, Centre for User Oriented IT Design

Nada, Dept. Computing Science

KTH, Royal Institute of Technology

S-100 44 Stockholm, Sweden

telephone: + 46 8 790 91 00

fax: + 46 8 790 90 99

e-mail: cid@nada.kth.se

URL: <http://www.nada.kth.se/cid/>

Sammanfattning

Rapporten behandlar frågan hur man skulle kunna åstadkomma en rikstäckande, distribuerad dokumentbas för skolbruk. Slutsatsen är att databasen bör bestå av flera lokala dokumentbaser som kopplas samman med databasprotokollet Z39.50. Lagringsformatet bör vara SGML med några vanliga standardiserade DTD:er som t.ex. TEI och HTML.

Rapporten går igenom hur databaser för dokumenthantering fungerar. På grundval av denna genomgång presenteras ett förslag till ett tänkt system samt hur det bör fungera del för del.

Inom examensarbetet gjordes en prototyp till systemet som användes vid ett antal användarstudier vilkas syfte var att ta reda på mer om lärares behov av och attityder till datoriserad informationssökning.

Användarstudierna visar att det finns ett klart behov av material specialskrivet för lärare. Det kan röra sig om lektions- eller laborationsidéer, det kan vara rena läromedel och det kan vara mer pedagogiskt inriktat material. Vidare gav användarstudierna idéer om vilka egenskaper ett sökgränssnitt till systemet bör ha.

ELECTRONIC DOCUMENT REPOSITORIES AS SUPPORT FOR TEACHERS

– A PROTOTYPE OF A NATIONWIDE, DISTRIBUTED

ELECTRONIC DOCUMENT REPOSITORY FOR USE IN SCHOOLS

Summary

The report deals with the question of how to accomplish a nationwide, distributed electronic document repository for use within the educational system. The conclusion of the report is that the repository should consist of several local repositories interconnected with the database protocol Z39.50. The documents should be stored in SGML using a few common standardized DTD:s like TEI and HTML.

The report features a review of the workings of databases for document storage. Based on this review a hypothetical system is proposed and presented.

As part of the project a prototype of the proposed system was created that was used in a number of user studies. The purpose of these studies was to find out more about teachers' needs of, and attitudes towards, computerized information searching.

The user studies show that there is a clear need for a repository of written material created specifically for teachers – written material such as ideas for lessons or experiments, finished educational material, purely pedagogical material, and so on. Furthermore the user studies resulted in some ideas of what properties a user interface for the system should have.

INNEHÅLL

Inledning	7
Dokumentbaser behövs i skolan	7
Ett distribuerat system	8
Materialet i basen bör vara SGML-märkt	8
Syftet med examensarbetet	8
Metod och uppdelning av arbetet	9
Om dokumentbaser och söksystem	10
Om kapitlets innehåll	10
Allmänt om dokumentbaser	10
Om forskningsområdet	11
Om användargränssnitt och sökning	11
Teknisk uppbyggnad av sökfunktion	13
Z39.50	14
WAIS	15
Om metadata	16
Om lagring och lagringsformat	17
Lagringsformatet bör vara rikt	17
Lagringsformatet bör vara en öppen standard	17
Formatet bör vara innehållsbeskrivande	18
SGML är bäst lämpat	18
Om HTML	19
HTML används presentationsbeskrivande	19
HTML som lagringsformat	19
Om hur valet av DTD påverkar författaren	20
Inläring	20
Verktyg	20
Slutsats	21
Om Utmatning och konvertering	21
DSSSL	21
Konvertering med översättningsfilter	22
Att läsa på skärm	22
Om konverteringsstrategier	22
Förutseende konvertering	23
Konvertering på begäran	23
Det tänkta systemet	24
Introduktion	24
Den fysiska arkitekturen i det tänkta systemet	24
Systemet bör vara robust	24
Systemet bör ha flera nivåer och vara hierarkiskt	24
Det skall vara enkelt att lägga till material	25
Systemet bör bygga på TCP/IP och Z39.50	25
Dokumenterna i basen	26
Gränssnittet	26
Sammanfattningar och metadata ställer krav på dokumenten	27
Om konvertering	27

Prototypens uppbyggnad	28
Om kapitlets innehåll	28
Som lagringsformat valdes TEI och TEI-Lite	28
Om TEI	28
Varför valdes TEI?	28
Hur påverkas författaren?	29
Databasen	30
Funktioner i Free WAIS-sf	30
Hur Free WAIS-sf fungerar	31
Om Perl-modulen WAIS.pm	31
Användargränssnittet	32
Användargränssnittet version 2	33
Användargränssnittet version 3	34
Programmeringstekniska detaljer	37
Att sätta upp en textdatabas med Free WAIS-sf	37
Att bygga ett webb/CGI-gränssnitt	37
Hur fungerar prototypens CGI-program	38
Uppmärksningen av dokumenten i basen	39
Syftet med uppmärksningen	39
Om den märkning som behövs för prototypen	39
Märkning av en text för engelskundervisning med TEI	39
Märkning av en text för fysiklärare med TEI-Lite	41
 Användarstudier	 42
Om användarstudierna	42
Om kapitlet	42
Förutsättningar	43
Tillvägagångssätt	43
Försökspersoner	44
Studenterna	44
Lärarna	44
Lärarytbildarna	44
Två omgångar i studierna	45
Om presentationen av resultatet	45
Presentation av resultat	45
Datorer	45
Skolarbete	47
Prototypen	48
Övriga Synpunkter	52
Slutsatser och konsekvenser för det föreslagna systemet	54
Om lärares behov i stort	54
Om prototypens sökgränssnitt	54
Lärares behov kontra gratisprogram och standarder	55
 Bilaga 1	 57
 Litteraturförteckning	 61

INLEDNING

DOKUMENTBASER BEHÖVS I SKOLAN

Det har hänt mycket med det svenska skolsystemet de senaste åren – såväl i grundskolan som i gymnasiet. I och med de omfattande reformerna och övergången till målstyrning får de enskilda skolorna allt större ansvar för den övergripande planeringen av undervisningen. Detta ställer i sin tur krav på att lärarna och skolorna har tillgång till läromedel som är anpassade för de egna, lokala behoven. Det blir allt svårare för en skola att låsa sig vid ett specifikt läromedelspaket för flera år i taget. Ännu svårare blir det att ha ett litet antal läromedelspaket som skall fungera för alla skolor i hela landet. För att hjälpa lärare och skolor att uppfylla de nya krav som ställs krävs nya arbetsverktyg.

Broady (1992) presenterar i en artikel *Kunskapsverkstaden, lokala dokumentbaser som arbetsverktyg för lärare* ett antal krav man bör ställa på bra arbetsverktyg för lärare. Utgångspunkten är en analys av lärares behov. Bland de behov som behöver stödjas ingår lärares behov av att framställa eget läromedels- och referensmaterial, deras behov av att på ett bra sätt kunna söka och hitta redan färdigt material att använda eller att utgå från, samt deras behov av att dela med sig och sprida det material som redan har framställts i olika syften. Lärare måste också ständigt fortbilda sig och följa med vad som händer inom sina olika ämnesområden och detta kräver i sin tur tillgång till kvalificerat faktamaterial [1].

En dokumentbas – en databas för dokumenthantering – med material för lärar- och skolbruk, där lärare själva lägger till nytt material de har framställt, skulle med säkerhet kunna vara en bit på vägen att uppfylla dessa krav och behov. Detta är en av slutsatserna i Broadys artikel och det har varit en av utgångspunkterna för examensarbetet.

Mycket litet har i övrigt skrivits om lärares behov av datorstöd. Från Skolverket är utgångspunkten att informationsteknologi skall användas i undervisningen men man har litet att säga om hur informationsteknologi bör användas i övriga delar av lärararbetet. Andersson (1997) har i ett examensarbete varit med och utvärderat ett projekt där både lärare och elever arbetade helt papperslöst och där Internet användes som enda informationskälla. Resultatet av utvärderingen visade att Internet för närvarande inte kan ersätta bra läroböcker av flera skäl. Två tydliga brister var bristen på struktur på webben och bristen på specialskrivet material. Dock kan Internet i många fall vara ett bra komplement till traditionella läroböcker [2]. En specialutformad dokumentbas med material skrivet av lärare för andra lärare skulle däremot kunna bidra till att lösa båda dessa problem.

Ett distribuerat system

I och med att Internet växer och blir allt mer spritt blir det allt lättare att åstadkomma distribuerade system för en mängd ändamål. För dokumentbaser är fördelarna med distribution uppenbara: i stället för att ha ett antal fristående lokala dokumentbaser kopplar man ihop dem och får tillgång till den samlade informationsmängden. För skolor innebär det att man kan hämta material från skolor i hela landet och även att exempelvis Skolverket kan publicera material som blir åtkomligt för alla Sveriges skolor. Ett rikstäckande system för hantering av lärarproducerat material bör vara distribuerat.

Materialet i basen bör vara SGML-märkt

För att inte låsa sig vid en viss mjukvarutillverkare bör man använda ett lagringsformat som är en vedertagen, öppen standard. SGML är en sådan standard. Med SGML får man inga problem med inkompatibla ordbehandlingsformat eller bekymmer med olika länders teckenuppsättningar. SGML medger dessutom en mycket rik innehållsbeskrivande märkning av de dokument som lagras vilket gör att de kan vidarebehandlas på en mängd olika sätt t.ex. konverteras till andra format för utmatning. För mer information om SGML och dess egenskaper som lagringsformat se kapitlet *Om lagring och lagringsformat*.

SYFTET MED EXAMENSARBETET

Syftet med examensarbetet är att utreda hur de tekniska delarna av ett system för att hantera lärarmaterial kan se ut om man använder sig av relevanta, existerande, öppna standarder samt att göra en prototyp av systemet med enkla, allmänt tillgängliga verktyg. Prototypen syftar till att visa hur en del av systemet skulle kunna se ut, och därmed tjäna som utgångspunkt vid fortsatt utveckling. En viktig poäng med prototypen är också att visa att systemet faktiskt är realiserbart.

En liten serie användarstudier har genomförts där prototypen tjänade som utgångspunkt för att samla in mer information om lärares särskilda behov: hur materialet bör katalogiseras, vilka metadata som är viktiga att ha med, vilken typ av material som främst är intressant o.s.v. Andra syften är att ta reda på attityder till den här typen av system samt hur lärare hanterar arbetsmaterial idag.

Examensarbetet behandlar endast mycket ytligt de problem som rör författandet av SGML-dokument: vilka verktyg som bör användas, hur dessa verktyg bör se ut, hur mycket man som författare måste lära sig, hur man bäst gör det o.s.v. Detta problemområde är intressant i sig men det finns inte så mycket forskning gjord på området som man skulle önska. Det är däremot ett område som borde utredas eftersom det är viktigt för användbarheten av nästan alla SGML-baserade system.

METOD OCH UPPDELNING AV ARBETET

Dokumentbaser för att hantera lärarproducerat arbetsmaterial har knappast använts alls i svenska skolor. Hur ett rikstäckande system för att hantera lärarproducerat material skulle utformas var alltså mycket öppen fråga. Det naturliga tillvägagångssättet hade varit att försöka ta reda på mer om lärares behov genom att ta kontakt med ett litet antal lärare och genom intervjuer försöka samla in mer information. Denna möjlighet stod dessvärre inte till buds eftersom jag påbörjade examensarbetet ett par veckor före sommarlovet början.

Jag valde i stället att börja med att bygga en prototyp baserad på de krav och önskemål som redan fanns insamlade. Detta skulle göra att jag fick någonting konkret att utgå ifrån vid senare användarstudier. Tanken var att möjliggöra fortsatt iterativ design med mer detaljerade krav och synpunkter från lärare.

För att genomföra examensarbetet var jag tvungen att sätta mig in i en rad olika områden:

- ◆ SGML, XML, DSSSL, TEI, ISO 12083 och andra SGML-relaterade standarder samt vilka verktyg som finns att tillgå.
- ◆ Forskning om digitala bibliotek och digital litteratur samt de försöksprojekt som har genomförts i samband med detta.
- ◆ Verktyg och standarder för att bygga dokumentbaser: Z39.50, WAIS, Harvest.
- ◆ Mjukvara som behövdes för att bygga prototypen: CGI, Perl, free WAIS-sf, WAIS.pm.

Från denna fördjupning gick det att dra en hel del slutsatser om hur den tekniska sidan av systemet borde fungera. Ett ramverk för hur systemet kunde se ut skisserades. Utifrån detta byggde jag sedan en prototyp. Nödvändiga steg för att bygga prototypen var:

- ◆ Märka upp några testdokument att använda i prototypen.
- ◆ Sätta upp en databas innehållande testdokumenten.
- ◆ Bygga ett gränssnitt mot databasen med hjälp av webb- och CGI-teknik.

När en fungerande prototyp väl var klar fortsatte arbetet med att få tag på lämpliga personer som kunde pröva prototypen och ha synpunkter på den så att den kunde förfinas. Detta arbete tog en del tid; lärare är ett tidspressat släkte. Jag fick till slut tag på tio personer som var intresserade av att pröva prototypen och ha åsikter om den samt att ställa upp på en intervju. När användarstudierna var gjorda försökte jag sammanställa resultaten i tre användarkategorier: lärarstuderande, lärare och lärarutbildare. Jag försökte också sammanställa de insamlade synpunkterna på sökgränssnittet och gjorde en ny version av sökgränssnittet som skulle motsvara lärares behov bättre än den första versionen.

OM DOKUMENTBASER OCH SÖKSYSTEM

OM KAPITLETS INNEHÅLL

Det här kapitlet handlar om dokumentbaser och söksystem. Kapitlet tar framförallt upp den tekniska sidan av dessa system men även några olika aspekter på användargränssnitt och användbarhet. Kapitlet kan ses som en översikt och en bakgrund mot vilken resten av rapporten kan ställas.

ALLMÄNT OM DOKUMENTBASER

I takt med att Internet växer och blir alltmer tillgängligt och i takt med att datorer i nätverk blir allt vanligare blir det också allt vanligare med databaser för lagring, hantering och sökning av digitala dokument. Detta tycks bero på ett genuint behov av att kunna organisera, finna och sprida information – ett behov som kan återfinnas i en mängd olika delar av samhället. Det talas ofta om det nya informationssamhället men att "kunskap är makt" visste Francis Bacon redan på 1500-talet. Behovet av att organisera och sprida information är betydligt äldre än datorns historia.

Datoriserade dokumentbaser har använts länge inom flera delar av näringslivet och inom stora organisationer, exempelvis inom förlagsbranschen för att automatisera bokframställningsprocessen, i stora företag för att hantera rapporter och manualer och inom universitetsvärlden för att hantera stora mängder forskningsrapporter så att de blir sökbara och så att man kan göra papperskopior efter behov.

Även om dokumentbaser har en lång historia är det först på senare år, i och med den snabba utvecklingen av teknik för datorkommunikation, som det har blivit möjligt att göra dokumentbaser enkelt åtkomliga för stora mängder människor. Internets genombrott och utbredningen av TCP/IP har möjliggjort bra datorkommunikation till mycket rimliga priser.

Det är också först nu som teknik och program för att sätta upp och underhålla dokumentbaser blivit allmänt tillgänglig. Fortfarande är de bästa programmen mycket dyra och riktar sig till stora företag med massiva resurser, men gratisprogrammen blir allt bättre och är ibland sämre endast i rent handhavandemässiga avseenden; gratisprogram finns ofta bara till vissa operativsystem, behöver ofta kompileras och är ofta trassligare att installera, konfigurera och så vidare.

Möjligheterna att sätta upp egna, billiga system för avancerad dokumenthantering och sökning har aldrig varit större för den som har gott handlag med datorer.

OM FORSKNINGSSOMRÅDET

Forskning om dokumentbaser och digital litteratur kallas ofta för *Digital libraries research* och börjar växa fram som ett eget forskningsfält, även om det än så länge har mycket oskarpa gränser. Denna forskning grundar sig till en del på erfarenheter från försöksprojekt med att märka upp digital litteratur och göra den tillgänglig över Internet, ofta i första hand med lingvister och litteraturvetare som tänkta användare. En annan och oftast mer teknisk del av forskningsområdet handlar om publicering av forskningsrapporter på Internet. Från den här forskningen kommer flera olika tekniskt inriktade system för sökning och överföring av filer och dokument som FTP, WAIS, Harvest m.fl. Annan forskning rör t.ex. informationssökning, klassificering av digitala resurser och metadata, se avsnitten *Om användargränssnitt och sökning* respektive *Om metadata* längre fram i detta kapitel. I samband med denna forskning utvecklas ofta användbar mjukvara som blir fritt tillgänglig, om än i befintligt skick.

Tyvärr finns det finns mycket lite vetenskapligt material skrivet som behandlar ämnet på ett övergripande plan. Det mesta som skrivs är rapporter från olika projekt som nästan alltid har praktisk inriktning eller rör något litet delproblem. Den intresserade som vill försöka göra sig en bild av fältet rekommenderas att ta en titt på *D-lib Magazine* [3] som är en webbtidskrift som ges ut som en del i NSF/DARPA/NASA Digital Libraries Initiative – ett stort nationellt amerikanskt forskningsprogram.

OM ANVÄNDARGRÄNSSNITT OCH SÖKNING

Möjligheten att söka i en stor mängd dokument är ofta det viktigaste skälet till att man över huvud taget sätter upp en dokumentbas. Sökfunktionen i en dokumentbas är dessutom ofta den enda del av en dokumentbas som en användare ser. Därför börjar genomgången av dokumentbasers uppbyggnad med en genomgång av sökgränssnitt.

World Wide Web är idag ett av de största användningsområdena för Internet. Dess spridning är så stor att många datoranvändare felaktigt sätter likhetstecken mellan Internet och World Wide Web. Det är ett tydligt exempel på att användargränssnittet i hög grad formar människors uppfattning av ett system. Eftersom man vill underlätta för användare att använda sin tidigare kunskap kan det

alltså vara bra att se till att ens söksystem är försett med ett användargränssnitt på webben. På det viset kan det stora flertalet användare med någon datorvana använda systemet utan allt för hög inlärningströskel. Dokumentbasens innehåll blir också tillgängligt för alla med möjlighet till webbuppkoppling.

Man bör försöka gömma så mycket som möjligt av systemets interna konstruktion för användaren. Detta är webben mycket bra på. Man kan låta användaren formulera en sökfråga på en webbsida, men söka i flera olika databaser som kan ligga på olika maskiner och på olika platser [4].

Bara det faktum att ett system har ett webbgränssnitt mot användaren räcker inte för att systemet skall vara väl användar-anpassat. Till att börja med är informationssökning omöjlig att göra hur enkel som helst. Det är en färdighet som måste tränas upp. Bara att formulera vad man är ute efter i termer av sökord eller sökfraser kan vara nog så svårt för de flesta människor som kanske aldrig har tänkt i dessa banor tidigare [5]. Även om man har någon tidigare vana av söksystem, är de som finns på webben i dag sällan så enkla eller intuitiva att använda som man skulle önska. Nyckelfunktioner i frågesyntaxen göms undan i bruksanvisningar eller "expertsidor" som kan vara svåra att hitta. Det finns heller ingen samstämmighet om vad en given söksträng betyder mellan de olika sökmotorerna på webben i dag, Shneiderman, Byrd och Croft (1997), delvis beroende på att de olika sökmotorerna nästan alltid har speciella finesser de är ensamma om.

Shneiderman, Byrd och Croft påpekar särskilt vikten av tydlighet i gränssnittet. Det ska tydligt gå att se vilka möjligheter gränssnittet erbjuder. Användaren skall heller inte behöva lära sig något speciellt sökspråk för varje ny sökmotor. Allt som går att göra skall vara överblickbart och åtkomligt i användargränssnittet [6].

För att åstadkomma ett riktigt bra sökgränssnitt är det dock mycket att tänka på och mycket som kan gå snett. Man bör alltså inte tumma på det användarorienterade synsättet. Utvärdering med hjälp av användare och iterativ design är förmodligen omistliga i sammanhanget. Man bör också fundera över hur man ska tillgodose både nybörjarens och den erfarna informationssökarens behov, eventuellt genom att ha olika användargränssnitt för olika användarkategorier.

Shneiderman, Byrd och Croft delar in sökning i fyra faser:

1) Formulering

Det första man måste göra är att formulera sin fråga i termer av söksträngar och operatorer eller i vissa fall i termer av kända texter. Den här delen kan underlättas avsevärt med ett bra gränssnitt där man ser vilka kombinationer av söksträngar som är möjliga vilka fält som finns att söka på o.s.v. Själva tankearbetet med att reda ut exakt vad det är man letar efter kan man nog tyvärr inte hjälpa till med men man kan alltid vara tydlig med vad som går att göra.

2) Genomförande

Hur ska användaren tala om att steg ett är färdigt? Oftast görs det med en knapp men det är viktigt att det som står på den är tydligt och inte kan missförstås. En knapp med texten "sök" kan till exempel ofta vara en knapp som leder till en söksida snarare än en knapp som sätter igång en sökning. Det finns också system där sökningen sker direkt och uppdateras kontinuerligt utan att använ-

daren behöver ge något kommando för det. Denna variant är bra därför att den är blir mer interaktiv men det kräver mycket snabba databaser, något som är svårt när man har att göra med mycket stora databaser över webbresurser och där allting användaren gör måste sändas över nätet.

3) Granskning

Granskningsfasen är en mycket viktig del av sökningsarbetet. Vid sökning i stora databaser kan antalet träffar bli överväldigande. Om den enda metoden att gallra ut ytterligare är att titta på dokumenten ett och ett, är sökningen värdelös. Om man däremot kan få hjälp med att se vad dokumentet innehåller på ett kompakt och effektivt sätt är mycket vunnet. Här finns det en mängd tekniker för hur man kan göra, alltifrån att visa sammandrag, "abstracts", till visualiseringar av dokumentsrymder där man kan se hur dokumenten förhåller sig till varandra och till sökorden. Den vanligaste metoden är att visa upp titlarna på dokumenten plus de första raderna (Lycos m.fl.). Andra system visar abstracts (t.ex. Excite) och låter användaren styra i vilken ordning dokumenten skall presenteras (Altavista advanced query). Generellt kan sägas att den här delen är ett eftersatt men viktigt område.

4) Förfining

I dagens system kan användaren ofta ge respons på vilka dokument som var bra genom att ange exempelvis "more like this". Det här är ett bra sätt att hjälpa användaren att hitta rätt. Ett annat vanligt sätt är att låta användaren söka vidare på bara de dokument som frågan resulterade i. På det här området har man generellt kommit ganska långt. Det främsta som saknas är mer interaktiva system med snabba svarstider där användaren kan få återkoppling direkt vid en förfining eller förändring av frågan. Med tanke på storleken som behövs på databaser över webb-resurser och det ständigt ökande antalet användare lär det dröja innan hastigheterna kommer upp i nivåer som gör det möjligt. Dessutom stöder inte HTTP-protokollet den här typen av kontinuerlig kommunikation.

TEKNISK UPPBYGGNAD AV SÖKFUNKTION

För att kunna göra ett antal datorlagrade dokument sökbara behövs ett index och en sökmotor. Indexet skapas vanligtvis av ett program som är specialskrivet för att indexera digitala dokument. Indexeraren går igenom dokumenten ord för ord och skapar en förteckning över vilka ord som förekommer i vilka dokument, hur ofta de förekommer och var. Bland kommersiella sökmotorer är det även vanligt att man försöker att hitta vanliga fraser, ord som ofta förekommer tillsammans m.m. Mycket vanliga ord, som förekommer så ofta i dokumenten att det inte säger något om dokumentets innehåll, tas bort. Indexen varierar en hel del i storlek beroende på hur man gör. Man indexerar heller inte alltid hela dokumenten: indexeraren till Lycos indexerar t.ex. bara de första 20 % av varje dokument [7]. Sökmotorn är ofta beroende av att indexet ser ut på ett visst sätt eftersom vissa

typer av sökningar utnyttjar speciell indexinformation. Resultatet är att sökmotorn och indexeraren ofta hör intimt samman och säljs eller distribueras som ett paket.

Nuförtiden finns en stor uppsättning program för att sätta upp söksystem. En hel del bra mjukvara är dessutom gratis. Dessvärre är den mesta mjukvaran specialskriven för att hantera HTML-dokument, något som inte alltid är vad man vill ha. Vidare gör inte gratisprogrammen något med dokumenten – de skapar index av ett antal lagrade filer och arbetar med indexet. Somliga kommersiella system delar upp de lagrade filerna i delar och tillåter olika kombinationer av delarna att utgöra olika dokument. Jag kommer i rapporten endast att uppehålla mig vid funktionen hos gratisprogrammen. Jag ska också presentera två standarder för söksystem: Z39.50 och WAIS.

Z39.50

Z39.50 är en ANSI/NISO-standard som specificerar datastrukturer för att skicka sökfrågor från en klientmaskin till en databasserver och för att skicka sökresultat tillbaka. Z39.50 är alltså ett protokoll för att söka i dokumentbaser. Z39.50 säger ingenting om hur en databas skall se ut inuti, den specificerar bara hur databasen skall kommunicera och vilka frågor den skall förstå. Z39.50 är tillståndsberoende och hanterar sådant som sessioner, inloggningar och tidigare sökresultat som alltså kan bearbetas vidare efter en sökning. Z39.50 specificerar också ett antal frågesyntaxer som protokollet hanterar. Z39.50 bygger på ISO:s OSI-modell och är ett protokoll på applikationsnivå. De flesta implementationerna har gjorts ovanpå TCP/IP. Frågesyntaxen i Z39.50 medger en rad olika typer av sökfrågor allt från enkla sökord kombinerade med booleska operatörer till sökning på fält och på ord som förekommer i närheten av varandra.

Namnet kommer sig av att Z39 var ANSI:s kommitté *ANSI-accredited standards developing organization serving libraries, publishing and information services* och Z39.50 är den femtionde NISO standarden. Den senaste versionen av Z39.50 antogs 1995 och benämns ibland "version 3". Innan dess kom "version 2" 1992 och "version 1" 1988.

Z39.50 används oftast i samband med mycket stora dokumentbaser. Bibliotek världen över har ofta Z39.50-koppling mot omvärlden – då oftast bara till bibliotekets katalog. Den stora svenska katalogen Libris som underhålls av Kungliga biblioteket har Z39.50-koppling liksom katalogerna hos flera stora amerikanska bibliotek. En del stora intranet-program hanterar sökningar via Z39.50.

En av Z39.50:s stora fördelar är att det är en öppen standard. Det medför att den är vanligt förekommande och att det finns olika implementationer med olika väl utbyggd funktionalitet och pris; man kan alltså välja den implementation som passar bäst. Det medför också att man har en standardiserad söksyntax. Det går alltså att söka i flera databaser samtidigt med en enda fråga och vara säker på att frågan tolkas likadant av alla servrar. Det här kan vara mycket användbart i flera olika sammanhang. Man kan t.ex. ha ett enda gränssnitt till flera olika baser, man kan också tänka sig olika

avancerade gränssnitt till olika användare men som söker i samma databaser. Det kan också vara bra om man vill distribuera sin databas utan att det syns; man kan ju ha ett användargränssnitt som gömmer alla de olika delarna. Man kan också göra standardiserade sökgränssnitt där varje användare kan ställa in vilka databaser som skall sökas i [8].

Priset för stor funktionalitet är som vanligt att mjukvaran blir komplex. Detta gör i sin tur att mjukvara för Z39.50 ännu så länge bara finns fritt tillgänglig till UNIX. I kommersiella databassystem finns standarden däremot implementerad i ett antal produkter för t.ex. Windows NT. Dessa är å andra sidan för det mesta mycket dyra per licens. För mera information om Z39.50 se *The Z39.50 Information Retrieval Standard – A Strategic View of Its Past, Present and Future* [9] eller för en mer teknisk behandling: *Z39.50 in a nutshell* [10].

Isite

Isite är ett programpaket som utvecklats av CNIDIR – Center for Networked Information Discovery and Retrieval. Isite är en implementation av Z39.50-standarden och innehåller ett program som gör kopplingen mellan Z39.50 och HTTP möjlig – en så kallad gateway. Isite är den mest kraftfulla mjukvaran för textdatabaser som finns att tillgå gratis men den är också mer komplex än de flesta andra. Det krävs en hel del arbete att anpassa programmet så att det uppfyller de krav man har på databasens funktion, användargränssnitt m.m. I gengäld får man tillgång till en stor mängd funktioner, en del stöd för SGML-hantering och sessionshantering som gör att man kan logga in på systemet för att kontrollera vilka som har tillgång till det. Isite skapar också ganska små och kompakta indexfiler.

Jag valde att inte använda Isite som grund för prototypens söksystem eftersom de nödvändiga anpassningarna av systemet skulle bli onödigt krångliga. För framtida bruk i liknande situationer torde det däremot vara värt att titta närmare på Isite speciellt eftersom det hela tiden sker utveckling av systemet. För mera information om Isite se [11].

WAIS

WAIS (Wide Area Information Services) är ett protokoll för textsökning som utgör en delmängd av ANSI-standarden Z39.50. WAIS är byggt på TCP/IP och har en tillståndslös klient/server-arkitektur. Man har alltså en server som svarar på frågor från klienter med hjälp av TCP/IP. Det går bra att skapa klienter med webbgränssnitt med hjälp av CGI-program. WAIS klarar av nästan alla funktioner man vill ha i en sökmotor, AND, OR, NOT, NEAR, fraser, jokertecken, felstavningar samt ord förekommande i ett visst fält som gör att det möjligt att ange att man t.ex. vill söka enbart i titeln på dokument. Eftersom WAIS är en standard och inte ett program finns det olika implementationer med olika välutvecklad funktionalitet.

WAIS utvecklades 1988 av Brewster Kahle m.fl. Z39.50 tjänade då främst som inspirationskälla. WAIS fungerar inte ihop med Z39.50-1988. En viktig skillnad från Z39.50 är att WAIS är ett tillståndslöst protokoll, men WAIS skilde sig också på andra vis som gjorde det

inkompatibelt med Z39.50-1988. I och med Z39.50-1992 hade man dock ändrat så mycket att WAIS faktiskt var en delmängd och i och med version 3 ingår WAIS som en så kallad profil tillsammans med ett antal andra system [9].

Till min prototyp valde jag en UNIX-implementation av WAIS som heter free WAIS-sf och som är den mest utvecklade WAIS-versionen som är fritt tillgänglig. Mer information om free WAIS-sf kan man hitta på [12].

Till PC finns WAIS inbyggt i en del kommersiella sökmotorer som då oftast implementerar någon variant av hela Z39.50-standarden (som WAIS alltså är en delmängd av).

OM METADATA

Det blir allt svårare att hitta bra material om ett givet område på webben. Med hjälp av bättre indexeringsalgoritmer försöker man kompensera för den växande mängden material. Problemet är att rent fritextbaserade index fungerar bäst för små textmängder inom ett givet ämnesområde [13]. När indexen växer blir det problem med sökresultat som är i det närmaste oöverblickbara samt med betydelseglidningar hos indexord mellan ämnesområden.

En möjlig lösning på problemet är att använda så kallad metadata, d.v.s. data om data, exempelvis information om innehållet i dokument. För en van biblioteksbesökare är det här ingen ny idé. I ett bibliotek står böcker klassificerade efter ämnesområde. Slår man i katalogen kan man söka på författare, ämne, titel o.s.v. På baksidan av böckerna kan man ofta hitta någon slags sammanfattning eller presentation av innehållet. På webben finns denna typ av information tillgänglig i mycket liten omfattning. Har man tillgång till metadata kan man dock göra bättre sökningar eftersom man kan söka på metadata som i vissa fall säger mer om innehållet än ett index. I andra fall kan man specificera vilket metadataelement man vill söka på – exempelvis Einstein som författare och inte bara som omnämnd i texten. Har man bra metadata kan också presentationen av träffar göras mer informativ genom att användaren får se de metadata som finns tillgängliga för sökresultaten.

Bestämmer man sig för att metadata är en bra idé inställer sig genast frågan: vilka metadata bör man ha? Frågan är inte lätt att svara på. En viktig avvägning måste till exempel göras av hur mycket metadata man skall ha om varje dokument. Mycket metadata ger bra möjlighet att avgöra vad ett dokument innehåller men mycket detaljerade metadata är arbetsamt att framställa och kräver specialkunskap. Inom biblioteksvärlden finns två olika skolor [8]: de som hävdar att detaljerade metadata är viktigt även på webben och de som hävdar att det viktigaste är att något system för metadata börjar användas överhuvud taget.

För att metadata skall fungera bra måste man försöka åstadkomma en standardiserad struktur så att så många som möjligt använder system som fungerar tillsammans. Det finns för närvarande flera olika föreslagna standarder för metadata:

MARC och TEI är två etablerade standarder som tillåter mycket omfattande metadata och täcker i princip alla behov man kan ha som bibliotekarie eller arkivarie. Av dessa format är MARC kanske det mest utbredda och används exempelvis i många universitetsbiblioteks katalogdatabaser. TEI används i många projekt vilkas mål är att publicera digital litteratur. Se avsnittet *Om TEI* i kapitlet *Om prototypens uppbyggnad*. Att skapa metadata i dessa format kan dessvärre ta ganska mycket tid och kräver en hel del förkunskaper.

Ett försök att göra det lättare att skapa metadata är *The Dublin Core Metadata Element Set*. Idén här är att standardisera ett litet antal metadata-element som de flesta "vanliga människor" skall kunna förstå. På det här viset får man inte lika mycket information via metadatat men förhoppningsvis kan systemet bli mer allmänt utbrett och börja användas rutinmässigt vid webbpublicering.

För mer information om metadata se t.ex. [13] och [14].

OM LAGRING OCH LAGRINGSFORMAT

Lagringsformatet bör vara rikt

Lagringsformatet är det format som ett dokument lagras på i en dokumentbas. Det finns en uppsjö av text- och ordbehandlingsformat att välja bland. Vilket eller vilka lagringsformat man bör välja är en fråga som har flera aspekter. Från söknings- och konverteringssynpunkt vill man ha en så rik och innehållsbeskrivande märkning som möjligt. Framför allt är det viktigt med en rik märkning om man skall lagra dokument under lång tid – kanske tiotals år, vilket praktiskt taget är en evighet i datorbranschen. Ju längre ett dokument skall sparas desto svårare är det att förutsäga alla dess framtida användningsområden. Om man lagrar dokumentet i ett för torftigt format som bara räcker till de behov man har för tillfället kan märkningen behöva bättras på, eller göras om helt, i efterhand.

Lagringsformatet bör vara en öppen standard

För att undvika att hamna i beroendeställning till en viss programvarutillverkare bör man hålla sig till ett format som är en öppen standard. På det viset gör man det troligt att det finns mjukvara att tillgå från olika tillverkare och för olika hårdvaruplattformar dessutom ofta som gratisprogram. Genom att hålla sig till en öppen standard kan man också ofta försäkra sig om att det finns en tillräckligt stor skara användare för att mjukvara skall finnas tillgänglig i många år, något som inte alls är självklart när det gäller enskilda mjukvaruföretag och deras produkter.

Formatet bör vara innehållsbeskrivande

De flesta lagringsformat som är allmänt utbredda lagrar i första hand information om utseendet hos det lagrade dokumentet: vad det skall vara för typsnitt, var det skall vara fetstil etc. Det är precis vad som behövs för ordbehandling i grafisk, direktmanipulativ stil. Det är emellertid inte den enda typen av information om ett dokument som man kan tänkas vilja lagra. Mycket information i ett vanligt dokument ligger på en semantisk nivå som en dator inte kan hantera eller ens komma åt. Det kan röra sig om författarnamn, adresser, telefonnummer, blandning av språk m.m. För att medge effektiv datorbehandling måste man märka upp innehållet så att informationen kan utvinnas på syntaktisk nivå. All information som man vill att datorn skall kunna använda måste då märkas upp i lagringsformatet. Detta gäller särskilt metadata.

SGML är bäst lämpat

För närvarande är det SGML som bäst klarar av att lagra innehållslig märkning för skiftande ändamål. SGML står för Standard Generalized Markup Language och är en ISO-standard med beteckningen: ISO 8879:1986. SGML är ett språk för att beskriva dokumentstrukturer och för att märka information. SGML är en öppen standard, vilket betyder att det finns ett antal programvarutillverkare som tillhandahåller mjukvara till olika datorplattformar. SGML är också en väl mogen och etablerad standard som har prövats i en mängd sammanhang.

En viktig sak att påpeka är att SGML inte är något lagringsformat i sig själv; SGML är ett system för att skapa lagringsformat. SGML använder något som kallas Document Type Definition (DTD) som beskriver hur en viss typ av dokument skall se ut. Varje SGML-dokument måste ange en DTD som dokumentet håller sig till för att vara giltigt. DTD:n anger vad som kan märkas upp i formatet och när det får göras. SGML i förening med en viss DTD definierar alltså ett lagrings- eller dokumentformat.

Den logiska uppbyggnaden i ett SGML-märkt dokument anges med hjälp av märkord som talar om hur olika delar av ett dokument skall tolkas. Text som är uppmärkt med ett visst märkord kallas ett *element*. För en bok vill man kanske märka upp vem som är författare, vem som är utgivare, tryckår, upphovsrättsinformation, titel, kapitel, rubriker o.s.v. För detta behöver man en lämplig DTD. DTD:n talar om vilka märkord som får förekomma i dokumentet, hur de får kombineras m.m. En stor fördel med SGML är att man själv kan bestämma vad som skall kunna märkas upp genom att skapa egna dokumenttypsdefinitioner (DTD:er).

Att skriva en DTD kan tyvärr vara krångligt och tidskrävande men som tur är finns ett antal standardiserade DTD:er för olika användningsområden som man oftast gör klokt i att använda. Dessa är oftast väl testade och genomtänkta. Till vissa av dem finns det till och med specialskriven mjukvara för exempelvis konvertering till andra format. När man väljer DTD måste man göra en viktig avvägning för att hitta en DTD som medger tillräckligt informationsrik märkning och som samtidigt inte är för komplicerad.

SGML är en mycket stor och komplicerad standard. Det är därför inte möjligt att ge annat än mycket summarisk information om

SGML i denna rapport. För den som vill veta mera finns det dock ett stort utbud av introduktioner av varierande omfattning. Lämpliga utgångspunkter kan t.ex. vara: [15], [16], [17] eller [18].

OM HTML

HTML används presentationsbeskrivande

HTML är idag den mest använda SGML DTD:n. Ändå är det så att de flesta som skapar dokument i HTML inte har en aning om vad SGML är och än mindre att HTML skulle vara en applikation av det. En viktig bidragande orsak är att HTML inte används som det ursprungligen var tänkt; HTML används i stor utsträckning idag till grafisk och typografisk utformning. Detta gör att många som tillverkar webbsidor använder märkorden enbart för att åstadkomma ett visst utseende på sin text. Man utnyttjar att ett visst element får ett visst, standardiserat utseende i webbläsaren i stället för att använda märkorden som de var tänkta: för att beskriva struktur och innehåll i dokumentet och överlåta presentationen åt webbläsaren.

Ett exempel på det är HTML:s tabeller som idag för det mesta används för att positionera text på en sida, sätta text i spalter etc. Det finns också en rad grafiska verktyg för att skapa webbsidor på WYSIWYG-maner och den HTML-märkning som dessa verktyg genererar är ofta inte det minsta strukturbeskrivande. Allt detta gör att idén med struktur- och innehållsbeskrivande märkning har gått förlorad i HTML-världen.

En annan orsak till att HTML-användare inte känner till SGML är att webb-läsare över lag är så förlåtande att man inte behöver skapa ett korrekt HTML-dokument för att en webb-läsare skall kunna visa dokumentet – man behöver alltså nästan inte kunna någon SGML-syntax alls.

HTML som lagringsformat

Som HTML används idag (som presentationsbeskrivande format) är det ett tämligen uselt lagringsformat. Däremot, om HTML-DTD:n används som den var tänkt från början och märkorden används för att beskriva strukturen hos ett textdokument, blir det betydligt mer intressant. HTML-DTD:n är visserligen enkel men den räcker till för att märka upp strukturen i enkla rapporter och hel del annat. HTML innehåller också ett system för att kunna märka upp enkla metadata. Problemet består i att få människor att börja använda HTML på ett mer struktur- och innehållsbeskrivande sätt. Ett steg i rätt riktning har tagits i och med introduktionen av formatmallar till HTML som gör att presentationen kan kopplas fri från de enskilda märkorden. Det återstår dock att se om den här idén får tillräckligt genomslag. För mer information om formatmallar se *Utmatning och konvertering*.

OM HUR VALET AV DTD PÅVERKAR FÖRFATTAREN

Inlärning

Valet av DTD får konsekvenser för den som skall skapa dokument. För att skapa ett dokument i ett format som är inriktat på innehållslig märkning krävs ett visst mått av kunskap om formatet. Ju rikare märkning formatet tillåter, desto fler märkord innehåller det och desto mer är det att hålla reda på. För en person som jobbar med ett komplicerat format ofta och regelbundet går det nog bra, men den höga inlärningströskeln kan ställa till problem för den som bara skall skapa dokument någon gång i bland.

En del grundläggande kunskaper om märkord och hur de fungerar måste en person som skall framställa SGML-dokument ha. Hur mycket SGML personen behöver kunna beror på vilka verktyg som finns att tillgå. Det finns grafiska verktyg som gör att författaren inte behöver kunna någon SGML-syntax alls men man måste fortfarande veta en del om märkning och märkord. SGML tillåter inte att märkorden placeras hur som helst. Hur de får användas beskrivs i DTD:n. Vissa element får kanske bara förekomma inuti vissa andra element. Ett exempel kan vara att ett postnummer bara får förekomma i en adress. Detta styrs av DTD:n och en annan DTD kan mycket väl tillåta postnummer var som helst i dokumentet, något som kan vålla bryderi hos en användare som inte kan någonting om SGML.

Verktyg

Det finns ett antal så kallade användarvänliga, grafiska verktyg för att hjälpa till med författandet av SGML-dokument. Ett stort problem är att den stora spännvidden mellan olika dokumenttyper gör det omöjligt att åstadkomma ett optimalt användargränssnitt för alla olika DTD:er; en DTD för matematiska formler kräver förmodligen en annan sorts författarverktyg än en DTD för romaner och så vidare.

Den vanliga typen av generellt författarverktyg börjar med att ladda in DTD:n och sedan lägga upp märkorden i menyer. Bara märkord som är giltiga i den aktuella kontexten är valbara från menyn. Ett problem med det här är att en användare ofta inte vet vad de olika märkorden betyder. Ofta har DTD:erna obskyra förkortningar som märkord vilket gör att man inte är helt hjälpt trots att man kan se vilka märkord som är giltiga för tillfället. Att märkordet <Q> ofta står för citat är till exempel inte helt självklart för en svensk nybörjare. Här skulle man kunna dra nytta av ett verktyg som är specialskrivet för en viss DTD eller som åtminstone går att anpassa och ställa in efter användarens behov. De generella SGML-verktyg som finns idag lämnar en hel del övrigt att önska vad gäller möjlighet till användaranpassning.

Verktyg som är specialskrivna för speciella DTD:er har än så länge lyst med sin frånvaro. Den enda DTD som har förärats egna författarverktyg är HTML-DTD:n (HTML är ju en tillämpning av SGML). Detta beror till största delen på att det är dyrt att skriva speciellt anpassade verktyg och att man därför måste se till att ha ett stort antal potentiella kunder. HTML är en så vitt utbredd DTD att

kundunderlaget är stort. Andra SGML-DTD:er är inte tillnärmelsevis så utbredda. Det kommer nog dröja innan det dyker upp specialskrivna verktyg till fler DTD:er.

Slutsats

Det tycks vara ofrånkomligt att ställa ett visst mått av extra krav på författaren om man vill använda SGML och innehållsbeskrivande märkning i sitt lagringsformat. Förutom att skriva den rena texten måste författaren ange de olika dokumentdelarnas funktion i form av märkord som bestäms av DTD:n. Detta innebär ett merarbete som bara är försvarligt då den utökade funktionaliteten verkligen behövs. Detaljerad innehållsbeskrivande märkning kommer därför inte att ersätta utseendeorienterad HTML när det gäller snabb och enkel webbpublicering med små krav på sökbarhet och där den förväntade livslängden på dokumentet är kort. Däremot, när det gäller dokument som skall underlätta mer kvalificerad användning är det viktigt att dokumenten har den information som krävs för möjliggöra bra sökningar, bra uppskattningar av dokumentets innehåll och bra datorbearbetningar. Än så länge är någon form av SGML det enda standardiserade sättet att göra det på. Det gäller då att försöka uppskatta hur detaljerad märkningen behöver vara så man väljer en DTD som inte ställer högre krav på författaren än vad som behövs.

OM UTMATNING OCH KONVERTERING

Om materialet i dokumentbasen är lagrat i SGML i ett hanterbart antal olika DTD:er kan man erbjuda utmatning i olika presentationsinriktade format. RTF, TeX, HTML osv. Ett krav är att man gör en utseendebeskrivning av något slag, en så kallad formatmall (eng. stylesheet), för varje DTD som förekommer i basen. Förslagsvis använder man då DSSSL som är en nyligen framtagen standard för detta ändamål. För själva konverteringen använder man sedan ett program som läser SGML och DSSSL och som skapar en ny fil i det önskade formatet som stämmer med den angivna formatmallen så långt det är möjligt i det aktuella utmatningsformatet. Dessbättre är det ofta viktigare att få utmatning av textinnehållet i rätt format än att få utmatning som exakt överensstämmer med formatmallen.

DSSSL

DSSSL (uttalas *disse*l) står för: Document Style Semantics and Specification Language och är ISO-standard 10179:1996. DSSSL är ett språk för att beskriva typografisk utformning av SGML-dokument och är tänkt att komplettera SGML. Standarden är dess värre relativt ny, så det finns inte så många implementationer än. I skrivande stund är *James Clarks DSSSL Engine – JADE* den enda implementationen som är gratis. JADE klarar av att ge utmatning i RTF och

TeX och kan översätta från en DTD till en annan. För mera information om DSSSL se t.ex. [19].

Konvertering med översättningsfilter

Innan DSSSL fanns var (och är till stor del fortfarande) översättningsfilter det vanliga sättet att göra konverteringar på. Metoden går ut på att man skriver ett program med instruktioner för hur man ska konvertera från en viss SGML-DTD till ett annat format. Man kör filtret med en SGML-fil som indata och får ut en ny fil i det önskade formatet. Det finns en del verktyg som är specialskrivna för den här typen av uppgift. Vad som behövs är i regel avancerad strängmatchning och strängersättning.

Den här metoden innebär att man måste skriva en separat rutin för varje par av DTD och utmatningsformat. Antalet konverteringsrutiner som behöver skrivas blir då lika med antalet utmatningsformat gånger antalet lagringsformat – en funktion med kvadratisk komplexitet – vilket gör att man tvingas hålla antalet olika lagringsformat och utmatningsformat nere till ett hanterbart minimum. Med formatmallar är man friare.

Vilket konverteringsverktyg man bör välja beror till stor del på vad man trivs med. Paket för att skriva konverteringsrutiner finns till Perl: SGMLSPM och till TCL/TK: COST. Information och mjukvara finns att hämta på [20] resp [21]. Information om andra konverteringsprogram finns att hämta från *The Whirlwind Guide to SGML Tools and Vendors* [22].

Att läsa på skärm

Ett alternativ till att konvertera dokumentet är att presentera det på skärm i en SGML- eller HTML-läsare. Än så länge finns ingen SGML-läsare som läser DSSSL men man kan nästan alltid skapa egna formatmallar för utseendet – som dock inte är kompatibla mellan de olika SGML-läsarna. Nu finns också SGML-läsare som insticksmodul till HTML-läsare vilket gör att man kan ha samma program för att läsa både HTML och SGML.

OM KONVERTERINGSSTRATEGIER

En viktig teknisk fråga är huruvida konvertering till de olika utmatningsformaten skall ske en gång för alla när ett dokument läggs in i basen, eller på begäran, då en användare vill ha ett visst dokument i ett visst format.

Det här är en fråga som har flera aspekter och ett entydigt svar på vilken metod som är bäst i alla lägen går inte att ge. I första hand handlar det om att välja mellan två olika administrativa strategier: "förutseende konvertering" eller "konvertering på begäran". För en

utförligare genomgång än den som ges här, se *Just-in-time conversion just-in-case collections* [23].

Förutseende konvertering

”Förutseende konvertering” innebär att man lagrar konverterade versioner av alla ursprungsdokument i databasen ”för att vara förutseende”. För slutanvändaren betyder naturligtvis färdigkonverterade filer att överföringstiden kortas ner och följaktligen en bättre service.

En viktig faktor att beakta är att komplexiteten hos administrationen påverkas. Ett databassystem som är inte gjort för att hantera samma dokument i flera olika format kräver att någon manuellt håller reda på hur de olika versionerna lagras, hur de namnges o.s.v. Proceduren att lägga till nya dokument kompliceras genom att den som lägger in dokument måste konvertera varje nytt dokument till alla de olika formaten och lagra en fil för varje format. Skall någon ändra i ett dokument måste filer för alla de olika utmatningsformaten omskapas. Det kan också hända att utmatningsformat ändras och kommer i nya versioner.

HTML är ett exempel där nya versioner dyker upp med bara några månaders mellanrum. Om man då vill utnyttja de senaste tilläggen måste man generera om alla dokument i basen med en ny konverteringsrutin vilket kan vara ett jättearbete. Det finns å andra sidan kommersiella dokumenthanteringssystem som automatiserar det mesta av den administrativa hanteringen.

Konvertering på begäran

Konvertering ”på begäran” innebär att man bara lagrar dokument i ett format och konverterar till andra format på begäran. Eftersom denna strategi innebär att användaren får vänta på konverteringen innebär konvertering ”på begäran” en långsammare service än ”förutseende konvertering”. Från den administrativa sidan är ”konvertering på begäran” mycket enklare att hantera, speciellt för stora dokumentbaser där många dokument begärs fram mycket sällan och där vissa utmatningsformat bara används av ett fåtal personer.

För att lägga till ett nytt dokument med den här strategin behöver man bara lägga till en fil i databasen. Sedan tar systemet hand om resten. Om det exempelvis kommer en ny HTML-version behöver man bara aktualisera konverteringsrutinen. Vill man lägga till ett nytt lagringsformat behöver man bara se till att det finns en formatmall för formatet eller andra konverteringsrutiner.

Eftersom konvertering av ett dokument går ganska fort jämfört med överföringstiden behöver inte den extra tiden som konverteringen tar märkas så mycket för användaren men valet av strategi innebär en avvägning som är värd att fundera över.

DET TÄNKTA SYSTEMET

INTRODUKTION

Med utgångspunkt från innehållet i förra kapitlet kan man dra en mängd slutsatser om hur ett rikstäckande system för dokumenthantering i skolan bör fungera. Nedan följer en översiktlig redogörelse för hur jag anser att systemet bör fungera – del för del. I vissa fall går det att bygga beslut på dokumenterade erfarenheter från andra projekt och på det som tidigare tagits upp i rapporten, men i andra fall är det mina egna åsikter som förs fram.

DEN FYSISKA ARKITEKTUREN I DET TÄNKTA SYSTEMET

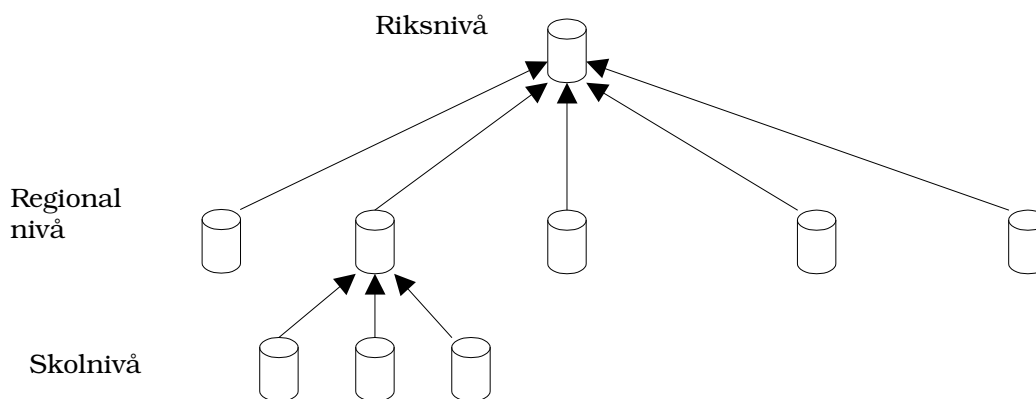
Systemet bör vara robust

Robusthet är en mycket viktig egenskap systemet bör ha. Systemet skall fungera även om delar läggs till eller tas bort. Speciellt viktigt är det här i ett uppbyggnadsskede då olika skolor måste kunna komma i gång i den takt de själva finner rimlig; systemet måste få växa fram i sin egen takt. Detta kan göras genom att varje skola får sätta upp sin egen dokumentbas – enligt vissa riktlinjer – som sedan kan knytas ihop med andra, med samma struktur, till ett större system som fortfarande inte behöver vara tätare sammanknutet än man önskar.

Systemet bör ha flera nivåer och vara hierarkiskt

I en ickehierarkisk, distribuerad databas finns det inget centrum. Man kan i princip se det som ett antal separata databaser. För att göra en sökning måste man hålla reda på alla delarna och skicka sökfrågor till var och en. Detta är svårt att genomföra i praktiken men även om det är tekniskt möjligt så blir kommunikationen i systemet lidande. Om meddelanden skickas ut till alla delar av systemet varje gång en sökning görs blir det en mycket stor mängd meddelanden. Det är inte en bra lösning.

För att lösa problemet kan man dela in systemet i flera nivåer och låta det få formen av ett träd där varje löv representerar en lagringsplats eller en lokal dokumentbas. Noderna innehåller index över dokument på underliggande nivåer samt egna dokument om så önskas. Se figur 1. Riksnivån är roten i trädet. Sedan har man undernivåer som kommun eller region och slutligen de enskilda skolorna. Sökning kan göras i de delar som ligger på vägen från den lägsta, lokala noden till roten. Det betyder att det som finns inlagt i en viss nod bara är åtkomligt för de noder som är nedanför den. Man



Figur 1. Principskiss av det tänkta systemet.

kan också tänka sig att antalet undernivåer till en viss nod kan variera och styrs av lokala behov. Att lägga upp en ny nivå påverkar bara de noder som är direkt under den – förutsatt att de kan komma överens påverkas ingen annan. Tanken är att noderna på högre nivå skall utgöra en slags samlingsbaser som innehåller information om det som finns i alla baserna på nivån under i trädet. Vid varje nivå görs dock ett urval – mycket material som läggs upp på en skola är förmodligen inte intressant för andra skolor eller regioner.

En fördel med en sådan här lösning är att en skola som vill lägga upp en dokumentbas bara behöver hålla reda på de noder som är rakt ovanför i trädet. En annan fördel är att skolorna eller regionerna kan ha dokument upplagda som inte är allmänt åtkomliga – om varje nod kan ange vad som skall föras vidare uppåt i trädet blir vissa dokument bara åtkomliga från de noder som finns nedanför i trädet. Detta gör att en skola kan lägga upp material som man inte vill sprida utan som bara görs åtkomligt på skolan.

Det skall vara enkelt att lägga till material

Det bör vara mycket enkelt att lägga upp dokument så att de blir sökbara och allmänt åtkomliga. I princip bör det bara behövas ett "handgrepp", som t.ex. att spara dokumentet på rätt ställe. Någon del av systemet skall sedan ta hand om att indexera och lägga in det nya dokumentet i basen, exempelvis genom en nattlig körning en gång om dygnet. Dokumentet skall sedan vara sökbart och åtkomligt för alla på samma nivå och under. Det är alltså viktigt att ha bra gränssnitt och eller hjälpprogram som automatiserar så stor del av publiceringsprocessen som möjligt.

Systemet bör bygga på TCP/IP och Z39.50

För att de olika lokala dokumentbaserna skall kunna kommunicera behöver man ett gemensamt protokoll. ANSI/NISO Z39.50 eller dess lillebror WAIS torde vara precis vad som behövs. Mjukvara finns dessutom både i form av mycket utvecklade kommersiella system och i form av enklare men fullt fungerande system som är fritt tillgängliga.

För den grundläggande, mer fysiska delen, av kommunikationen faller det sig naturligt att använda Internet och TCP/IP som kommunikationsmedium av flera skäl:

- ◆ Internet är billigt, väl utbyggt och allmänt förekommande.
- ◆ En mängd program, tekniker och kommunikationsprotokoll är utvecklade för Internet-bruk. Tillsammans utgör de en svårersättlig kombination.
- ◆ World Wide Web har snabbt etablerat sig i människors medvetande. Man bör kunna utnyttja webbens möjligheter i systemet.

Faktum är att Internet i dag har en så oomstridd ställning att det är svårt att tänka sig en annan lösning även om det är tekniskt möjligt. För att använda Internet som kommunikationsmedium krävs att de serverdatorer som har hand om de lokala dokumentbaserna har fasta förbindelser med Internet.

DOKUMENTEN I BASEN

Dokumentet i basen bör vara SGML-dokument i ett begränsat, hanterbart antal DTD:er. Kravet på de DTD:er som används är att de medger tillräckligt rik märkning vad gäller metadata för att bra sökningar skall kunna göras och att det finns formatmallar för konvertering och utmatning.

Man måste komma överens om vilka fält som det skall gå att söka på i dokumentbasen så att man kan söka i hela systemet med en enda sökfråga. Detta ställer i sin tur krav på vilka DTD:er som går att använda. För att ta ett exempel: Om man bestämmer sig för att det skall gå att söka på nyckelord måste den DTD man använder medge uppmärkning av nyckelord. Alla DTD:er som medger sådan märkning går å andra sidan i princip att använda. Här är det nödvändigt att ta reda på vilken information som användarna upplever som viktig att kunna söka på innan man låser sig vid en viss uppsättning.

GRÄNSSNITTET

Sökverktyget till dokumentbasen bör i första hand vara en webbapplikation, dels för att göra systemet lättåtkomligt och dels för att kunna erbjuda ett enkelt och bekant gränssnitt åt de flesta användare. Med webb och CGI-teknik går det dessutom lätt att bygga flera olika gränssnitt för olika kategorier användare.

I sökfunktionen skall det gå att söka dokument på alla nivåer i systemet i samma sökning. Det skall också gå att tydligt se vilka möjligheter sökgränssnittet erbjuder användaren. För allmänna krav och synpunkter på sökgränssnitt se avsnittet *Om användargränssnitt och sökning* i förra kapitlet.

Sammanfattningar och metadata ställer krav på dokumenten

Man vill kunna se någon form av kortfattade presentationer av de dokument som en sökning resulterar i innan man börjar läsa eller ladda ett enskilt dokument. Den här funktionen finns t.ex. redan i Skolverkets databas Länkskafferiet [24]. När man har gjort en sökning här visas först korta beskrivningar av dokumenten, sedan väljer man de dokument man vill se i sin helhet. Den stora skillnaden mot det system jag föreslår är att länkskafferiets beskrivningar är inskrivna för hand av särskilda redaktörer medan beskrivningarna i det system som jag föreslår genereras automatiskt genom att utnyttja SGML-märkningen i de lagrade dokumenten.

Förutsättningarna för den här typen av presentationer och metadata är olika beroende på vilka typer av dokument som används i dokumentbasen. Ett TEI-dokument har exempelvis ett omfattande dokumenthuvud som man kan använda, ISO 12083-dokument har ett speciellt abstract-element o.s.v. Ju färre olika typer av dokument basen innehåller ju lättare är det att hantera. Möjlighet till bra metadata ställer egentligen de hårdaste kraven på vilka, och hur många olika DTD:er som går att hantera. Att man kan få en bra presentation av sina sökresultat är å andra sidan en mycket viktig hjälp vid sökning i stora dokumentbaser; den här aspekten måste alltså läggas stor vikt vid. Se avsnittet *Om användargränssnitt* i förra kapitlet.

Naturligtvis bör utseendet och strukturen på presentationerna vara lika för alla dokument i dokumentbasen. Detta gör att inte alla DTD:er kan användas. De DTD:er som används bör dessutom ha liknande strukturer för sammanfattningar och metadata.

Om konvertering

Skolan är överlag en resursknapp organisation. Därför det är viktigt att systemet är så enkelt att administrera som möjligt. Då är det rimligt att konvertering av dokumenten sker på begäran i de flesta fall. Det gör också att det blir lättare att ha mycket stora dokumentbaser.

I och för sig finns det integrerade databaslösningar där konvertering i förväg är nästan lika enkelt att administrera som konvertering på begäran men den som väljer ett sådant system riskerar att binda sig väl hårt till just det systemet och den programtillverkaren.

En viktig sak att nämna är att alla delar av systemet inte behöver fungera likadant. Bortsett från väntetiden spelar det ingen roll för användaren om dokumenten konverteras i förväg eller på begäran – huvudsaken är att användaren i slutändan får sitt konverterade dokument. De olika delarna av systemet kan alltså använda den strategi som passar bäst i det enskilda fallet.

PROTOTYPENS UPPBYGGNAD

OM KAPITLET INNEHÅLL

I det här kapitlet presenteras den prototyp som har byggts i samband med examensarbetet. Prototypen är i form av en WAIS-databas med ett specialbyggt webbgränssnitt. Påpekas bör att även om systemet är fullt fungerande är det ganska klumpigt, krångligt att flytta, långsamt o.s.v. så det går inte att ta ut i verkligheten och använda direkt som det är. Prototypens uppbyggnad kommer att presenteras del för del.

Till prototypen SGML-märktes också två texter för att demonstrera SGML:s möjligheter för undervisnings- och lärarmaterial, en för engelska och en för fysik. Hur dessa texter valdes ut och märktes upp beskrivs i slutet av kapitlet.

SOM LAGRINGSFORMAT VALDES TEI OCH TEI LITE

Om TEI

TEI (utläses "tei") eller Text Encoding Initiative är ett projekt för att standardisera SGML-märkning av all slags litteratur och historiska källor för forskningsändamål. Det är ett omfattande projekt som har resulterat i en omfattande DTD och ett omfattande kompendium för att beskriva hur DTD:n bör användas.

TEI innehåller ett mycket stort antal märkord och ofta flera uppsättningar märkord för att märka upp samma typ av information. Detta är både en svaghet och en styrka. Svagheten ligger i att DTD:n blir mycket omfattande och komplex. Styrkan ligger i att man nästan alltid har tillgång till de märkord man kan tänkas behöva i en given situation.

TEI har slagit igenom på bred front inom området digital litteratur. Det finns t.ex. ett stort antal databaser med TEI-märkt engelsk litteratur tillgänglig på nätet se t.ex. [25]. För mer information om TEI se [26], [27] och [28].

Varför valdes TEI?

Det finns ett antal faktorer som avgjorde valet av DTD till TEI:s fördel:

- ◆ TEI medger en mycket rik märkning för humanistiska ändamål och tillåter sådant som grammatikalisk märkning av text vilket kan vara bra exempelvis i samband med språkundervisning.

- ◆ TEI & TEI Lite har likadana header-element och därigenom samma metadata.
- ◆ TEI:s & TEI Lite:s header är mycket stor och utförlig och innehåller det mesta man kan tänkas behöva av metadata.
- ◆ TEI & TEI Lite är enkla att bygga ut efter behov – exempelvis för matematiska formler.
- ◆ TEI är en mycket spridd och använd standard. Dessutom finns det en del fri mjukvara som är gjord för hantera just TEI.
- ◆ TEI Lite utgör en delmängd av TEI som använder samma header-element som TEI men annars bara använder de vanligaste elementen. TEI Lite är betydligt lättare att lära sig än TEI och har man lärt sig TEI Lite kan man ganska enkelt ta klivet över till den fullständiga TEI utan att behöva lära om någonting.

En viktig sak att påpeka är att TEI och TEI-Lite valdes som lagringsformat till prototypen i första hand för att visa på den funktionalitet som går att uppnå. Det behöver inte vara TEI och TEI-Lite som visar sig vara bäst för skol- och lärarbruk i slutändan, när alla relevanta faktorer har vägts in. Det viktiga är att lagringsformaten medger en tillräckligt informationsrik innehållslig märkning. Dessutom kan det komma att behövas specialiserade DTD:er för exempelvis lokala arbetsplaner.

Hur påverkas författaren?

Som beskrevs tidigare påverkar valet av lagringsformat de krav som med nödvändighet måste ställas på författaren. Med författare avses här en person som skall åstadkomma ett dokument i ett visst format även om texten redan är skriven, se avsnittet *Lagringsformatet påverkar författarens situation*. För TEI gäller detta alldeles speciellt. För att skapa dokument som håller sig till TEI-standarden krävs rätt mycket kunskap om formatet. Tröskeln för att jobba med TEI-Lite bör däremot inte vara oöverstiglig för de flesta förutsatt att man får en lämplig introduktion och har möjlighet till hjälp av någon som kan mer när man behöver.

Seaman (1997) beskriver i en artikel: *The user community as responsibility and resource – building a sustainable digital library* ett projekt med digital litteratur där systemutvecklare samtidigt agerat utbildare. Detta har resulterat i en mycket bra kontakt mellan utvecklare och användare, vilket kan var ovärderligt för att åstadkomma ett bra system [29]. Något liknande borde kunna fungera ihop med utveckling av det system som föreslås i den här rapporten.

TEI har fördelen att man kan ignorera det stora flertalet märkord. TEI är medvetet gjort så att det skall finnas ett överflöd av märkord så att man inte skall sakna något. En viktig sak att lära sig som nybörjare är hur man läser sig till hur olika märkord skall användas och hur man hittar märkord man inte har använt förut men som man plötsligt behöver. Detta gör man förslagsvis genom att slå upp i TEI-standarden som finns upplagd på nätet på [26].

Man skulle mycket väl kunna tänka sig ett lättlärt, användarvänligt, specialskrivet verktyg för att författa TEI-dokument men tyvärr har inget sådant dykt upp än så länge. En orsak är att TEI-formatet är komplicerat och att det därför innebär ett mycket stort arbete att

skriva ett sådant verktyg. Det är dessutom vanligt att man bygger ut TEI för olika ändamål och det måste verktyget också klara av att hantera. Förmodligen finns det dessutom inga stora pengar att tjäna på det heller eftersom TEI, än så länge, nästan uteslutande används i universitetsvärlden.

Utvecklingen av användarvänliga generella SGML-editorer sker dessbättre hela tiden. Den senaste trenden är att göra det möjligt att i hög grad anpassa dem efter författarens behov och preferenser. Med ett sådant verktyg skulle det alltså vara möjligt att i hög grad anpassa det till TEI-redigering. Tyvärr är dessa verktyg mycket dyra än så länge så problemet kvarstår till stor del.

En mycket viktig sak att hålla i minnet är att TEI och TEI-Lite bara är de format som valdes till prototypen. Det kan mycket väl vara så att användarstudier visar att andra format är bättre lämpade.

DATABASEN

Till prototypen valdes Free WAIS-sf V2.1 som databasprogram. Free WAIS-sf är ett fritt tillgängligt, UNIX-baserat programpaket för att skapa WAIS-databaser som har utvecklats vid universitetet i Dortmund. Valet gjordes av flera skäl: Ett mycket starkt skäl var att WAIS är en delmängd av Z39.50 standarden, d.v.s. söksyntaxen är standardiserad och universell. Man kan alltså söka i flera databaser med samma fråga och veta att den tolkas lika av alla de olika databaserna – förutsatt att de förstår WAIS. Ett annat skäl är att Free WAIS-sf har stöd för fält och fältbaserad sökning något som är nödvändigt om man vill kunna utnyttja SGML-märkningen i lagringsformatet till fullo. Till Free WAIS-sf finns också en Perl-modul som ger ett programmeringsgränssnitt mellan Perl och WAIS som gör att det är enkelt att utforma ett eget sökgränssnitt på webben med de funktioner man vill ha. Programmeringen görs i Perl i form utav ett CGI-program. Det fanns också ett krav på att den mjukvara som användes skulle vara mycket billig. Free WAIS-sf utgår ifrån redan lagrade filer och skapar index över dem, så all information som skall infogas i dokumentbasen måste finnas i dokumenten.

Funktioner i Free WAIS-sf

Free WAIS-sf hanterar sökning på ord och hela fraser. Orden och fraserna kan kombineras med de booleska operatorerna OCH, ELLER och ICKE. Man kan även ange >, < eller = för numeriska data. Free WAIS-sf hanterar även sökning på specifika fält som exempelvis författare eller titel. Det går också att använda synonymordlistor och plocka ut grundformer av ord som indexerats för minska indexens storlek och förbättra sökresultat. Allt som allt gör dessa funktioner Free WAIS-sf till ett mycket användbart verktyg även om det innehåller en del programfel.

Hur Free WAIS-sf fungerar

För att få Free WAIS-sf att fungera på ett bra sätt måste man skriva en formatbeskrivning för de dokument man vill indexera. I formatbeskrivningen anger man vilka fält som skall indexeras och hur de så kallade *WAIS headlines* skall se ut för dokumenten (vanligtvis titel och författare). *WAIS headlines* är den information som WAIS skickar tillbaka för de dokument som matchar en sökfråga. Free WAIS-sf skickar tillbaka träffarna i relevansordning – ungefär ”bästa träffen först”. Free WAIS-sf ger också relevanspoäng på träffarna för att man skall kunna samsortera träffar från flera olika baser.

För att programmet skall veta hur det skall urskilja de fält det skall indexera måste man tala om hur de ser ut. Det gör man med hjälp utav så kallade reguljära uttryck: ett för att beskriva vad som markerar början på ett fält och ett för att beskriva vad som markerar slutet. Allt som finns i dokumentet mellan start- och slututtrycket kommer att indexeras så som tillhörande det fältet. Ett enkelt exempel kan vara att ha <författare> som startuttryck och </författare> som slututtryck.

Ett problem med att ange fält med hjälp av reguljära uttryck är att det inte fungerar fullt ut med SGML. Med SGML kan man nämligen sätta element inuti element. Praktiskt betyder det att man kan ha märkordet <författare> i olika betydelser beroende på i vilket sammanhang eller vilken kontext det förekommer: man kan märka upp författare i en bibliografisk referens men man kan också märka upp författaren till den aktuella texten. Man vill förmodligen kunna skilja på dessa två element men de har båda samma märkord: <författare>. Genom att skriva smarta reguljära uttryck kan man ofta kringgå problemet i enskilda fall men till det generella problemet finns det ingen lösning. Tyvärr!

Problemet ligger i det indexeringsprogram som följer med Free WAIS-sf. Med ett annat indexeringsprogram som stödjer idén med ”fält i fält” skulle man kunna hantera SGML fullt ut men tyvärr finns inget Z39.50-kompatibelt program som gör det på något enkelt sätt och som är gratis. Isites programpaket är den mjukvara som klarar av det bäst men det kräver en del omprogrammering för att fungera.

För mera information om Free WAIS-sf se den manual som finns på webben [12].

OM PERL-MODULEN WAIS.PM

WAIS.pm är en Perl-modul (paket med funktioner) som innehåller ett programmeringsgränssnitt mot WAIS. Med den här modulen är det mycket enkelt att göra anrop till WAIS-databaser, något som är mycket praktiskt när man skall programmera användargränssnitt. Naturligtvis är det inte absolut nödvändigt med en sådan här modul, men det är mycket bekvämt. Modulen innehåller dessutom i princip en helt egen WAIS-klient så man kan göra sökningar utan att behöva anropa något separat program utanför Perl. För lokala sökningar innebär det dessutom att inga nätverksuppkopplingar behövs alls. Modulen WAIS.pm finns att hämta från CPAN [20].

ANVÄNDARGRÄNSSNITTET

Användargränssnittet till prototypen består av ett sökformulär i form av en webbsida samt ett CGI-program skrivet i Perl som hanterar inmatning från formuläret, gör om det till en WAIS-fråga, skickar frågan till de aktuella databaserna och därefter hanterar och presenterar resultatet. Valet att använda Perl för att skriva CGI-programmet grundade sig mest på att det skulle vara snabbt och enkelt att implementera samt att det skulle vara enkelt att göra ändringar. Att skriva det i C skulle förmodligen snabba upp det en del men programmeringen skulle ta ganska mycket längre tid. Eftersom programmet bara ingår i en prototyp var inte snabbhet den viktigaste egenskapen som eftersträvades.

Under examensarbetets gång har tre versioner av gränssnittet utformats. Den första var ett utkast som gjordes utan att jag hade samlat in någon särskild information från användarna. De funktioner som finns i den första versionen av användargränssnittet är:

- ◆ ord och strängsökning
- ◆ operatorerna OCH, ELLER och INTE
- ◆ möjlighet att söka på följande fält: titel, författare, ämnesklassificering/nyckelord, header (d.v.s. all metadata om dokumentet. Jag hoppades få förslag på vad fältet borde heta på svenska.) samt hela dokumentet.
- ◆ sökning i databaser på olika nivåer: skolan, i regionen eller i hela landet.

Gränssnittet består av tre inmatningsfält för sökfraser. En sökfras kan bestå av ett eller flera ord. "Albert Einstein" eller "fysiker" är båda exempel på sökfraser. Man söker på en eller flera sökfraser genom att mata in en sökfras i ett sökfäld. Man kan ange var i ett dokument systemet skall söka efter sökfrasen genom att välja ett fält i en kombinationsruta bredvid sökfäldet. T.ex. "Albert Einstein" som "författare". Skall man söka på mer än en sökfras måste man ange hur de skall kombineras. Detta gör man genom att klicka på radioknappen mellan sökfälden. Ett exempel kan vara att söka på "Albert Einstein" OCH "fysiker". Se Figur 2.

När svaret på en sökning presenteras, visas fem träffar i taget för att det inte skall bli för mycket på en gång. För att se de följande fem träffarna klickar man på en länk längst ner på sidan. De enskilda träffarna presenteras med ett utdrag från dokumentets TEI-header. De delar jag har valt att plocka ut är: distributör, författare, titel, project description, notes statement samt keywords. Prototypen presenterar informationen under rubrikerna Titel, Författare, Distributör och Nyckelord. Resten av informationen presenteras som en sammanfattning. Förhoppningsvis ger det här en hyggligt fyllig bild av vad dokumentet innehåller helt automatiskt. Se Figur 3.

För att titta på hela dokumentet klickar man på länken för utmatning i SGML som ännu så länge är det enda utmatningsformat som stöds av prototypen. Tanken är att det skall finnas fler länkar här för utmatning i olika format.

Det absolut enklaste sättet att göra sig en bra bild av prototypens användargränssnitt är att titta på den på webben: [30].



Figur 2. Sökformulär till den första versionens sökgränssnitt.



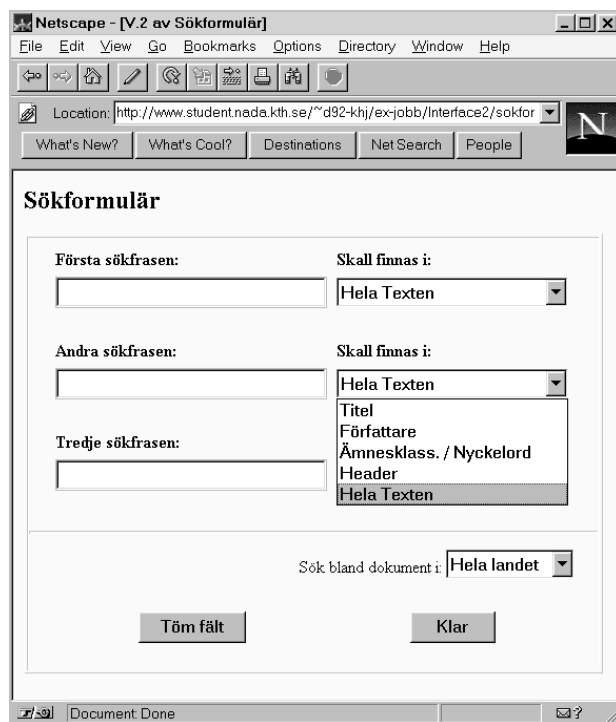
Figur 3. Presentation av sökresultat för en sökning på fysik.

Användargränssnittet version 2

De två första försökspersonerna som fick prova prototypen tyckte båda att möjligheten att ställa in vilken logisk koppling det skulle vara mellan sökfrasfälten gjorde sökningar mycket krångliga. Det

blev onödigt många val att ta ställning till. Båda försökspersonerna tyckte att det ändå bara var logisk och-koppling som var intressant. Se kapitlet *Användarstudier*.

Eftersom en stor del av dessa båda försök gick åt till att diskutera den här funktionen beslöt jag mig för att ta bort den till de andra användarstudierna. Om någon försöksperson skulle sakna möjligheten att ställa in kopplingen mellan sökfrasfält kunde jag visa version ett och sedan kunde vi diskutera skillnaderna. Version två [30] av användargränssnittet är identiskt med version ett förutom att version två har en fast logisk och-koppling mellan sökfrasfälten. Se figur 4.



Figur 4. Den andra versionen av sökformuläret till sökgränssnittet.

Användargränssnittet version 3

Version 3 [30] av användargränssnittet gjorde jag efter att ha avslutat användarstudierna. Det gällde att försöka väga samman alla synpunkter jag hade fått på de tidigare versionerna för att kunna åstadkomma ett användargränssnitt som i högre grad var anpassat efter lärares speciella behov än de båda tidigare versionerna. Se kapitlet *Användarstudier*.

Fältnamnen och rubriken ändrades. Rubriken blev "skall finnas" och fälten fick i tur och ordning heta: i titeln, som författare, som ämne, som nyckelord, som avsett användningsområde och någonstans i dokumentet.

En ny menyvalsruta lades till med rubriken Ungefärlig nivå. De valmöjligheter som finns där är åk. 1-3, åk. 4-6, åk. 7-9, gymnasiet och alla nivåer. Med den här rutans hjälp är det tänkt att användaren skall kunna ange vilken nivå på materialet som är intressant.

Den här funktionen stöds för närvarande inte av databasen eftersom implementationen skulle ta en hel del tid. Som det nu ser ut går det dock att göra sig en bra bild av hur användargränssnittet borde fungera, även om funktionen inte gör någonting i prototypen.

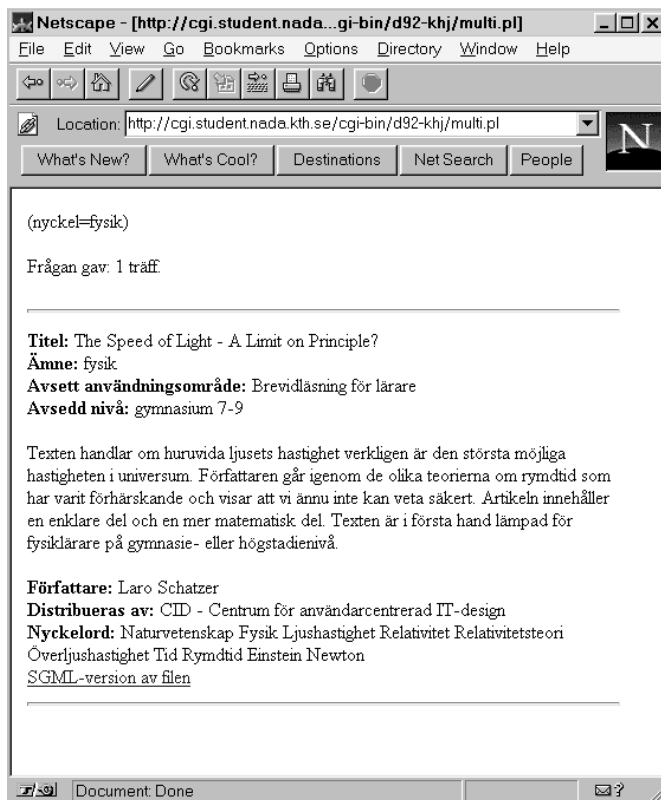
I den andra menyvalsrutan tog jag bort möjligheten att söka på Regionen eftersom flera försökspersoner ansåg att det var en helt onödig funktion. Skolan byttes ut mot den egna skolan för att förtydliga. Se Figur 5.

The screenshot shows a Netscape browser window with the title "[V.3 av Sökformulär]". The address bar shows the URL "http://www.student.nada.kth.se/~d92-khj/ex-jobb/Interface3/sokform.ht". The search form is titled "Sökformulär" and contains the following elements:

- Första sökfrasen:** An empty text input field.
- Skall finnas:** A dropdown menu with the selected option "någonstans i dokumentet".
- Andra sökfrasen:** An empty text input field.
- Skall finnas:** A dropdown menu with the selected option "någonstans i dokumentet". A list of search criteria is visible below the dropdown: "i titeln", "som författare", "som ämne", "som nyckelord", "som avsett användningsområde", and "någonstans i dokumentet".
- Tredje sökfrasen:** An empty text input field.
- Ungefärlig nivå:** A dropdown menu with the selected option "alla nivåer".
- Sök bland dokument i:** A dropdown menu with the selected option "Hela landet".
- Buttons:** "Töm fält" and "Starta sökning!".

Figur 5. Version 3 av sökformuläret.

Träffpresentationen ändrades också i ett par avseenden. *Ämne* har fått en egen rubrik. *Avsett användningsområde* har fått en egen rubrik. Det har även *Avsedd nivå* fått. Se Figur 6.



Figur 6. Presentation av sökresultat i version 3 av gränssnittet.

Ett tydligt problem var att många önskemål som olika försöks- personer hade inte går att uppfylla utan vidare med de gratisprogram som finns tillgängliga. En knapp "more like this" för att hitta liknande material är svår att stödja med hjälp av Z39.50. Länkar till relaterat material kan man däremot ha direkt i dokumenten. Att skapa en ämneskatalog med rubriker liknande YAHOO stöds inte av något gratisprogram som jag känner till som samtidigt stödjer Z39.50. Längden på texten uppgav flera försökspersoner att de ville kunna se. Det är dock inte så lätt att ange hur många sidor ett elektroniskt dokument upptar; det beror ju på storlek på teckensnitt m.m. Språk har jag medvetet avstått ifrån att ha en egen rubrik då den övervägande delen av materialet kommer att vara på svenska och en egen rubrik för språk skulle innebära ett slöseri med utrymme. Vilket språk en text är bör dock anges i sammanfattningen. Vilket datum texten lades in i systemet bör stå i texten, inte i sammanfattningen, av utrymmesskäl

Andra önskemål kräver metadata som inte stöds direkt i någon vanligt förekommande DTD. Detta kan naturligtvis lösas genom att bygga ut en befintlig DTD, men det skapar nya problem med spridning av den nya DTD:n, utbildning o.s.v. För att kunna märka upp "avsedd nivå", "ämne" och "avsett användningsområde" i prototypens dokument var jag tvungen att använda speciella konstruktioner i TEI som måste se ut på samma sätt i alla dokument i basen. Detta gör alltså att det blir mer att hålla reda på för den som vill publicera dokument i systemet. Problemet blir detsamma om man vill använda exempelvis HTML i databasen. Då måste man bestämma och stan-

andardisera vilka META-ID:n som skall användas för de olika typerna av metadata.

PROGRAMMERINGSTEKNISKA DETALJER

Att sätta upp en textdatabas med Free WAIS-sf

Att sätta upp en enkel databas med Free WAIS-sf är inte svårt. Man använder ett program som heter WAIS-index och anger bara vad databasen skall heta, var indexfilerna skall lagras samt vilka filer som skall indexeras. Sedan kan man söka på ord och fraser med hjälp av en godtycklig WAIS-klient. För att kunna använda fält i sina sökningar behövs en del arbete; man måste skriva en så kallad *format description file*. I denna fil anger man hur WAIS-index skall plocka ut fält. För varje fält som skall användas måste man ange ett start- och ett slututtryck. Ett enkelt exempel i SGML: för att skapa ett fält för titel anger man <TITLE> som startuttryck och </TITLE> som slututtryck (om det nu är så DTD:n ser ut). Allting som finns mellan märkorden <TITLE> och </TITLE> kommer nu att anses finnas i fältet. Använder man flera olika DTD:er får man skriva en *format description file* för varje DTD och lagra de olika dokumenttyperna i olika kataloger tillsammans med sina respektive formatbeskrivningsfiler.

Start och slututtrycken anges med reguljära uttryck. Reguljära uttryck är ett sätt att specificera klasser av strängar; ett reguljärt uttryck är ett strängmönster. Ett reguljärt uttryck motsvarar alltså alla strängar som stämmer överrens med mönstret. Reguljära uttryck är ett mycket kraftfullt sätt att arbeta med textsträngar. Dess värre är det, som nämndes tidigare, inte tillräckligt kraftfullt för att generellt kunna beskriva delar av SGML-dokument. Vad man får göra är att lusläsa DTD:n och försöka täcka in så mycket som möjligt med sitt uttryck.

Ytterligare ett problem består i att Free WAIS-sf bara matchar reguljära uttryck rad för rad. Så beroende på var ett SGML-dokument har radbrytningar (något som inte spelar någon roll för SGML) kan möjligheten att plocka ut fält korrekt helt försvinna. Lösningen är att redigera dokumenten en aning innan man lägger in dem, så att man ser till att det finns slutmärkord och man inte har radbrytningar där de inte får vara. I mitt fall räcker det att se till att TEI-headern uppfyller de här kraven eftersom alla de fält jag har använt kommer därifrån. D.v.s. inga radbrytningar i TEI-headern och inga borttagna slutmärkord får förekomma.

Att bygga ett webb/CGI-gränssnitt

WAIS-protokollet är till sin natur tillståndslöst. Det finns inga uppkopplingar eller sessioner. Man skickar en fråga och WAIS-servern skickar tillbaka ett svar. WAIS-servern har inget minne av vad den har gjort tidigare – på det sättet liknar WAIS-protokollet HTTP. Det

här gör att det är ganska lätt att skriva sina egna procedurer för att hantera WAIS men det behövs inte. Dels finns ett program som heter WAISQ som hanterar kommunikationen men där man måste skicka in ett korrekt frågepaket i ett speciellt format, dels finns en modul: WAIS.pm, till Perl som lägger till ytterligare en abstraktionsnivå, så att man har en funktion som tar två parametrar: själva frågan i klartext, samt en lista på de databaser man vill söka i. Funktionen skickar tillbaka svaren. Den här modulen användes i prototypen och har fungerat mycket bra.

För att bygga ett webbgränssnitt skriver man ett CGI-program som i princip består av fyra delar [31]:

- 1) Den första delen tar en fråga som indata från ett webbformulär och omvandlar den till korrekt WAIS-frågesyntax. Den här delen bör även ge vettiga felmeddelanden om frågan inte fungerar som användaren ställt den.
- 2) Den andra delen tar den nu korrekta frågan och skickar den till databasen. Detta görs enkelt med exempelvis WAIS.pm modulen om man programmerar Perl, annars kan man exempelvis använda UNIX-pipes och programmet WAISQ. Använder man WAIS.pm samsorteras träffar automatiskt om man söker parallellt i flera baser med samma fråga. Använder man WAISQ måste man göra det själv.
- 3) Den tredje delen tar hand om och presenterar sökresultatet.
- 4) Den fjärde och sista delen sköter utmatningen av de dokument som användaren vill se. Först måste dokumentet hämtas från basen. Det görs med fördel med WAIS.pm eller med WAISQ. Om dokumentet skall konverteras, gör man det nu. För det använder man några av de verktyg som omnämndes i avsnittet Om konvertering. Sedan skrivs dokumentet ut med lämpligt MIME-huvud beroende på vilket format som utmatningen sker i.

Hur fungerar prototypens CGI-program

CGI-programmet till prototypen följer ovanstående uppdelning. Den första delen av programmet omvandlar inmatningen från sökformuläret till en WAIS-fråga. Om inmatningen är konstig eller om inmatning saknas i något fält ges felmeddelanden som försöker tala om vad som är fel. Programkoden för den här delen är ganska trasslig och består av en stor mängd specialfall som måste hanteras.

Den andra delen av programmet använder WAIS.pm för att skicka en fråga till den valda databasen/databaserna.

Den tredje delen är den mest omfattande. För att kunna presentera bra information om varje dokument hämtas de in fem i taget och allt utom dokumenthuvudet kastas bort. Sedan används ett program som heter Sgrep för att plocka ut de relevanta delarna ur dokumenten med hjälp av SGML-märkningen. Sgrep är ett program som är gjort för att plocka ut delar av SGML-dokument och klarar av att plocka ut delar beroende på kontext (element inuti element). När man har plockat ut det som skall presenteras från varje dokument, skapar man en HTML-sida med länkar till filerna i sin helhet. För mer information om Sgrep se [32].

Den sista delen av programmet är inte särskilt märkvärdig eftersom prototypen bara erbjuder utmatning i SGML-format. Vad programmet gör är att anropa sig själv med det önskade dokumentets WAIS-id och sedan hämtas filen med hjälp av WAIS.pm och skrivs ut med ett SGML-MIME-huvud.

UPPMÄRKNINGEN AV DOKUMENTEN I BASEN

Syftet med uppmärkningen

Syftet med märkningen av de dokument som skulle läggas in i prototypbasen var att försöka ge en bild av hur SGML-märkt material speciellt framställt för lärare skulle kunna se ut. Eftersom olika lärarkategorier har olika behov valde jag att märka upp dokument för två ganska skilda kategorier lärare: fysik/NO-lärare och engelsklärare/språklärare. Jag märkte upp ett dokument för vardera kategorin. De dokument som är uppmärkta är till viss del ganska detaljerat märkta för att kunna visa vilka möjligheter som SGML erbjuder. I praktiken måste man här göra en avvägning mellan detaljnivån på märkningen och den tid det tar att åstadkomma den.

Om den märkning som behövs för prototypen

Prototypen använder ett element som heter *TEI-header* på olika sätt för att kunna stödja fältsökning. För att prototypen skall fungera bra krävs att "headern" innehåller en del märkning som inte är obligatorisk i TEI. Framför allt gäller det två element: nyckelord och ämnesklassificering. Dessa måste finnas med som elementen *keywords* och *textclass* inuti elementet *TEI-header*. För att få en användbar sammanfattning av dokumentet när sökresultat presenteras bör man ange *project description* och/eller *notes* i *notes statement*. För mer information om "TEI-headerns" beståndsdelar se TEI-standarden [26].

Märkning av en text för engelskundervisning med TEI

Textval och val av DTD

Valet av TEI som DTD för språkligt material föll sig naturligt; endast TEI i sin fullständiga form har verktyg för att göra ens en mycket enkel språklig analys som t.ex. analys av ordklasser. TEI:s standard för ordböcker verkade dessutom mycket intressant att titta närmare på för att se om den även skulle fungera för att märka upp en text för språkundervisning.

Den text som valdes som utgångspunkt var en TEI-märkt version av Aesopos fabler som finns på webben. Valet av text grundade sig på att de är välkända och på att de, trots sitt något ålderdomliga språk, är ganska lätta att förstå. Till skillnad från mycket av det som

är publicerat på nätet så består Aesopos fabler av ett flertal korta episoder som utan problem kan plockas ut ur sitt sammanhang. Dessutom är språket överlag enklare än i det mesta av den digitala litteratur på engelska som finns publicerad på Internet.

Det var två faktorer som främst styrde mig när jag gjorde urvalet av vilka fabler jag skulle märka. Den ena var svårighetsgraden på språket. Här var det ordval och språkets ålderdomlighet som främst spelade in. Den andra faktorn var hur välkänd fabeln är. Om fabeln är välkänd är den eventuellt lättare att läsa eftersom man vet ungefär vad som står. Detta medger att man kan ha ett något svårare språk.

Märkningen

Själva texten är märkt med vanliga nivåelement för att kunna ha rubriker i två nivåer samt textstycken. Svåra ord och fraser är märkta med gloselementet som finns i TEI. Dessa glosor/fraser är i sin tur länkade till gloslistan för varje fabel. Headern är uppmärkt för att fungera bra med prototypen och innehåller nyckelord och en kort innehållsbeskrivning i form av en "note". Den största delen av märkningen ligger dock i gloslistorna.

Gloslistorna innehåller mer komplicerad märkning i enlighet med TEI:s system för ordböcker. Denna märkning innehåller många olika möjligheter och är mycket flexibel. Det visade sig inte vara några som helst problem att använda den till avancerade gloslistor. Den märkning jag valde såg ut på följande sätt för varje glosa:

Först det engelska ordet eller frasen – ibland ordet på grundform och ibland inte beroende på vad som var lämpligt; ibland använde jag båda. Detta följdes av grammatikaliska bestämningar till ordet såsom ordklass, tempus, numerus, participform. I några fall angav jag även böjningsmönster för verb. Efter det kommer den svenska översättningen tillsammans med kommentarer om stilnivå, om översättningen är ungefärlig etc. Texten och dess märkning finns på: <http://www.student.nada.kth.se/~d92-khj/sgml/aesoptei.sgm>

Att skapa TEI-märkta texter

Att skapa en text som denna kräver en del olika typer av kunskaper. En är kunskap om TEI. Man behöver veta en del om vad man kan göra med TEI, hur TEI fungerar och vad de olika märkorden skall användas till. TEI tillhandahåller i många fall flera olika syntaktiska konstruktioner för att åstadkomma samma semantiska effekt. Man behöver alltså veta vilka märkord som hör ihop och vilka konstruktioner som är utbytbara. Man behöver en introduktion till TEI där man går igenom de viktigaste funktionerna samt tar upp konventioner som ofta används men som inte styrs av TEI i sig själv. Man behöver också med stor sannolikhet använda TEI:s riktlinjer att slå upp i under arbetet. Detta krävs mest i början men det finns nog ingen levande människa som kan hela TEI-standarden utantill. Har man väl kommit i gång är det inte så svårt men det finns en viss tröskel att komma över.

Märkning av en text för fysiklärare med TEI-Lite

Textval

Texten jag valde var en webbartikel av en fysiker i Schweiz som har en lite ovanlig syn på relativitetsteori. Valet av artikel var grundat på ett antal faktorer: artikeln skulle vara riktad till lärare snarare än elever; artikeln skulle innehålla en del formelhantering, artikeln skulle vara upphovsrättsligt fri; artikeln skulle vara hyggligt vetenskaplig och behandla ett område som lärs ut i skolan på en förenklad nivå.

Detta visade sig vara lättare sagt än gjort. Det tog lång tid att hitta en lämplig text. Att hitta genuint vetenskapliga artiklar var inte så lätt som jag trodde. Ett problem torde vara att formler inte fungerar bra i HTML. Att hitta enkla texter för elevbruk var enkelt men svenska elever får förmodligen problem med språket. Texten jag valde är inte perfekt men jag bedömde den som tillräckligt bra.

Valet av TEI-Lite föll sig rätt naturligt. TEI och TEI-Lite har ett bra dokumenthuvud och låter sig utvidgas för t.ex. formelhantering. Hela TEI DTD:n behövdes däremot inte varför TEI-Lite valdes.

Märkningen

Själva texten är märkt med vanliga nivåelement för att kunna ha rubriker i två nivåer samt textstycken. Fysikaliska termer är märkta med term-elementet som finns i TEI. Citat och bibliografiska referenser är också uppmärkta och referenserna i texten är länkade till litteraturlistan i slutet.

Bibliografin är ganska detaljerat uppmärkt; förmodligen mer än vad som vanligtvis behövs, men det var ett syfte i sig att utnyttja märkningsmöjligheterna så långt det gick. Något jag saknade var ett bra sätt att ange vilken tidskrift en viss artikel kom ifrån.

Vad beträffar formler hade jag tänkt att använda Mathematical Markup Language (MML) som tas fram av The World Wide Web Consortium. Tyvärr är inte MML en färdig standard ännu, därför består formlerna av GIF-bilder så länge. För mer information om matematisk SGML-märkning se [33] och [34].

Att arbeta med texten i undervisning

Texten är tänkt att fungera som inspiration/bredvidläsning för en fysiklärare på gymnasie- eller högstadienivå. Det är nog tveksamt om den kan användas i någon större utsträckning i undervisning, dels för att den kräver en del förkunskaper och dels eftersom den är på engelska, men det är inte omöjligt – det går ju att plocka ut delar.

ANVÄNDARSTUDIER

OM ANVÄNDARSTUDIERNA

Användarstudierna syftade till att vara lite mer än bara en utvärdering av den byggda prototypen. Genom intervjuer försökte jag också ta reda på så mycket jag kunde om lärares situation och om deras behov vad gäller tillgång till arbetsmaterial, faktamaterial, läromedel, tillgång till pedagogisk forskning och så vidare. Jag försökte också ta reda på hur lärare använder datorer i allmänhet och World Wide Web i synnerhet i sitt arbete idag. Hur mycket de samarbetar och hur de framställer och hanterar undervisningsrelaterat material var också av intresse.

Informationen jag samlade in var tänkt att ge en fylligare bild av lärares behov i de här avseendena än vad som fanns tillgänglig innan jag började med examensarbetet. Utvärderingen skulle också ge en uppfattning om vilket behov som finns av system liknande det som jag har skisserat samt vilka de viktigaste egenskaperna hos ett sådant system är sett ur användarnas perspektiv. Rena synpunkter på prototypens gränssnittsutformning skulle vidare ge mig möjlighet att förbättra det i en ny version.

Några statistiskt grundade slutsatser går inte att dra av användarstudierna eftersom antalet försökspersoner är för litet. Dessutom utgör försökspersonerna inte nödvändigtvis ett representativt urval. Studien är därför rent kvalitativ. För att ge en första inblick i lärares behov och situation torde det här tillvägagångssättet vara tillräckligt.

OM KAPITLET

Presentationen av användarstudierna och deras resultat nedan är ganska detaljerad. En del läsare kommer med säkerhet inte att tycka att alla detaljer är intressanta. Jag har dock valt att göra presentationen utförlig för den kategori läsare som vill ha alla detaljer och som vill kunna göra sig en egen bild av lärares situation så gott det går från mina användarstudier. De läsare som tycker att presentation blir långrandig uppmanas att helt enkelt hoppa framåt en bit och att inte läsa så noggrant. Resultatet av användarstudierna för prototypens del finns sammanfattat i kapitlet *Om prototypens uppbyggnad* i avsnittet *Användargränssnittet version 3*.

FÖRUTSÄTTNINGAR

Förutsättningarna för användarstudierna var i regel mycket primitiva. För att få tag på försökspersoner var jag tvungen att söka upp dem och träffa dem där de fann det praktiskt. För det mesta betydde det att jag fick söka upp dem på deras arbetsplats eller i deras hem. Ett minimikrav var dock att det fanns en dator med möjlighet till webbuppkoppling.

Tiden som gick åt till en studie varierade från 30 till 90 minuter. Någon direkt tidspress fanns bara i ett fall, annars berodde tidsvariationen mest på hur mycket försökspersonerna hade att säga under intervjudelarna (se nästa avsnitt). De primitiva förhållandena gjorde också att jag i de flesta fall inte hade möjlighet att spela in vad försökspersonerna sa utan fick anteckna så mycket jag hann med under försökens gång.

TILLVÄGAGÅNGSSÄTT

Studierna genomfördes med en försöksperson i taget. Varje studie började med att jag berättade lite om bakgrunden till den prototyp försökspersonen skulle få pröva, varför jag gjorde studierna o.s.v. Jag förklarade att jag hade gjort ett utkast till ett söksystem för lärarrelaterat material och behövde hjälp att förbättra det. Till utvärderingen hade jag en mall som jag använde med frågor jag absolut ville ha svar på. Dock ställde jag inte frågorna exakt som de står där utan försökte att få in dem naturligt under intervjuens gång. Jag ansåg det också viktigt att följa upp vissa av försökspersonernas svar med frågor som gick utanför mallen. Mallen jag använde kan ses i Bilaga 1. Studierna var upplagda på följande sätt:

Den första delen gick ut på att diskutera datorvana och datoranvändning i allmänhet. Till vad används datorn främst? Hur mycket? Har personen en egen dator? Sedan följde frågor om erfarenheter av Internet och World Wide Web samt åsikter om detta. Nästa del handlade om arbetet som lärare. Vilken typ av arbetsmaterial som används? Varifrån hämtas det? Varifrån får lärare idéer och pedagogisk inspiration? Hur arbetar de med att framställa material? Hur hanteras det sedan? Efter dessa frågor började själva utvärderingen.

Utvärdering av prototypen började med att försökspersonen fick se prototypens sökgränssnitt, sedan förklarade jag kortfattat hur det fungerade och försökspersonen fick ställa frågor. Försökspersonen fick nu i uppgift att söka rätt på en av de två texter som fanns uppmärksatta till prototypen. Det gick till så att jag beskrev i stora drag vad texten skulle handla om. Om försökspersonen undrade någonting under tiden var det bara att fråga. Vilken text försökspersonen blev tilldelad att söka på berodde på vilken typ av lärare försökspersonen var. I ett par fall var texterna ganska långt från något av försökspersonens ämnen. I dessa fall fick vi diskutera funktionen

och inte innehållet i databasen och försökspersonen fick låtsas att texten som denne skulle söka efter var relevant. Eftersom det ändå var funktionen hos prototypen som skulle utvärderas och diskuteras blev det här inget stort problem.

När försökspersonen fick träff på rätt text fick denne läsa träffpresentationen och ombads först att spontant tycka till om den. Sedan bad jag försökspersonen tala om hur han/hon skulle vilja ändra den om han/hon fick bestämma. Jag ställde också en del specifika frågor när de hade slut på synpunkter.

När vi gemensamt hade diskuterat träffpresentationen gick vi tillbaka till sökgränssnittet. Jag bad försökspersonen att pröva några mer komplicerade sökningar med fält och med flera sökfraser o.s.v. Sedan diskuterade vi sökgränssnittet på samma sätt som träffpresentationen.

När vi var klara med prototypen bad jag försökspersonerna ge sin syn på systemet – hur användbart det skulle vara, vilken typ av material som vore mest värdefullt, hur detaljerad märkning de fann önskvärd, hur de tänkte sig att publicering och användning skulle fungera eller inte fungera.

FÖRSÖKSPERSONER

Sammanlagt studerade jag tio försökspersoner. Av dessa tio var fem lärarstudierande i sista årskursen, två var praktiserande lärare och tre lärarutbildare.

Studenterna

En blivande gymnasielärare i samhällskunskap och religion.

En blivande gymnasielärare i svenska och historia.

En blivande gymnasielärare i latin och engelska.

En blivande gymnasielärare i engelska och tyska.

En blivande lärare åk. 4–9 i matematik och naturorientering.

Lärarna

En gymnasielärare i svenska och tyska.

En lärare åk. 1–7 i matematik och naturorientering.

Lärarutbildarna

En lärarutbildare i engelska på Institutionen för lärarutbildning, före detta kemilärare.

En lärarutbildare i datoranvändning, före detta yrkeslärare utbildande elektriker.

En doktorand på Institutet för lärarutbildning i Uppsala, före detta högstadielärare i naturorientering.

TVÅ OMGÅNGAR I STUDIERNA

De två första försökspersonerna hade båda mycket starka åsikter om sökgränssnittet till prototypen. De ansåg båda att möjligheten att ställa in vilka logiska operatörer som skulle kombinera sökfraserna var mycket förvirrande och onödig. Eftersom båda dessa försök kom att kretsa en hel del kring detta problem beslöt jag mig för att göra en ny version av gränssnittet till de övriga studierna. Denna version hade en fast logisk och-koppling mellanfälten och hade således något färre val för användaren att ta ställning till. Och-koppling mellanfälten valdes eftersom båda försökspersonerna uppgav att det var det enda som var intressant. Under de fortsatta studierna var det bara en person som saknade möjligheten att ställa in vilken typ av koppling det skulle vara mellanfälten.

OM PRESENTATIONEN AV RESULTATET

Eftersom de olika kategorierna av försökspersoner har lite olika utgångspunkter och bakgrund har jag valt att presentera de olika gruppernas åsikter och synpunkter var för sig.

PRESENTATION AV RESULTAT

Datorer

Datorvana och datoranvändning skiljer sig inte så mycket åt mellan de olika kategorierna. Det enskilt vanligaste användningsområdet är ordbehandling som är det användningsområde som alla uppger att de använder datorn till i första hand. Vad det är man skriver varierar dock mellan grupperna. Studenterna använder ordbehandlare till sina studier på olika sätt. Lärare framställer övningar, prov, klasslistor, brev etc. Lärarutbildarna skriver ofta mer forskningsrelaterat men skriver också till exempel material till sina kurser.

Det användningsområde för datorer som kommer på andra plats är förmodligen e-post som nästan alla har använt och som används oregelbundet men återkommande. På tredje plats kommer webben men det är inte mer än ungefär hälften av de som surfar som använder webben till arbetsrelaterade uppgifter. Andra användningsområden för datorer är uppslagsverk på CD-rom och kalkylprogram.

De flesta har egen dator men alla har åtminstone tillgång till dator.

Studenterna och webben

Hur mycket vana studenterna har av att använda webben varierar mycket; från knappt alls till ganska mycket. De som har surfat mycket har gjort det av eget intresse. De har knappt fått någon undervisning alls i hur man surfar och ingen alls i hur man kan använda webben som lärare i sitt ämne.

Trots att de inte har haft någon utbildning i att använda webben som lärare uppger dock alla att det finns tydliga problem med att använda webben som lärare:

- ◆ Det är svårt med källkritik på webben. Hur vet man vilken kvalitet det är på material som man hittar? Flera av studenterna uppgav att det är lätt att hitta åsikter av olika slag på webben, exempelvis politiska partiers eller enskilda personers uppfattningar, men det är svårare att hitta säkra fakta. För ämnen som samhällskunskap behöver det inte vara ett problem men i andra ämnen är det.
- ◆ Det är svårt att få en överblick över vad som finns på webben. Hur vet man om det är så att material verkligen inte finns eller om man bara inte har hittat det? Ämneskataloger kan vara till hjälp men de är inte alltid så bra och innehåller sällan så mycket material som man skulle önska.
- ◆ Det är svårt att göra bra sökningar. Antingen får man för många träffar eller också inga vettiga träffar alls. Det är också svårt att bedöma träffarnas kvalitet utan att gå igenom de resulterande webbsidorna. Allt detta gör att det tar lång tid att få fram information och material på webben enligt studenterna. För många ämnen som inte ändrar sig så mycket är det enklare och snabbare att slå upp i böcker.
- ◆ Det finns nästan inget material speciellt för lärare eller för skolbruk enligt flera av studenterna. Man skall ha enorm tur om man lyckas hitta någonting som handlar om det man söker efter på en lagom nivå, med en lagom omfattning och som dessutom riktar sig specifikt till lärare eller elever.

Lärarna och webben

Av de två lärare jag har pratat med varierar vanan av webben mycket. Den ena har surfat en del och den andra knappt alls. Även här är två problem tydliga: det är svårt att hitta på nätet och det tar lång tid. Att det tar lång tid beror till stor del just på problem med att hitta, men även på att det tar tid att värdera material för att bedöma svårighetsgraden och kvalitén på det. Sökmotorerna ger ofta alldeles för många träffar. Det är också svårt att hitta material som är bra.

Om det beror på att det bara är svårt att hitta eller om det är så att det de letar efter inte finns vet ingen av dem.

Lärarytbildarna och webben

Vana av webben varierar kraftigt inom gruppen, men alla har surfat mer än bara ett par gånger. Lärarytbildarna upplever att det inte finns så mycket material som man skulle önska som är skrivet för lärare eller för undervisning. Det är också ofta svårt att hitta material på lagom nivå och att anpassa materialet kräver för det mesta mycket arbete. Kvalitén på materialet varierar också kraftigt. Lärarytbildarna önskar också att det fanns mer forskningsinriktat, pedagogiskt material. Nyheter om pedagogiska forum, nya böcker o.s.v.

Skolarbete

Studenterna (som inte har så mycket erfarenhet, men lite...)

ATT ARBETA SOM LÄRARE

Enligt studenterna varierar det mycket från skola till skola hur mycket lärare samarbetar och delar material. Flera studenter har dessutom fått uppleva att lärare inte alltid är villiga att dela med sig av idéer och material. En del lärare anser att de har fått lägga ner mycket arbete på att framställa bra material till sin undervisning och har därför ingen lust att ge bort det hur som helst.

Pedagogiska idéer och inspiration anser studenterna att man främst får från andra lärare och från böcker. När man är nyutbildad så styrs dock ofta pedagogiken av det läromedel som används enligt flera av studenterna. Experimenterar gör man, enligt dessa, först när man har varit lärare ett antal år.

WEBBEN I LÄRARARBETET

Flera av studenterna anser att de får för lite undervisning i hur webben kan användas praktiskt i skolarbetet. De studenter jag har pratat med får en dags undervisning i det under hela utbildningen, uppdelat på ett par tillfällen.

Flera studenter menar också att Internet är olika bra för olika ämnen. Enligt dem är Internet till exempel ganska bra i samhällskunskap eftersom man dels ofta behöver aktuella fakta där, och dels ofta jobbar med åsikter, vilket aldrig är svårt att finna på Internet.

Det kan vara svårt med källkritik på webben. Flera studenter anser att webben är opålitlig och att det är svårt att veta vad man kan lita på. För att få fakta är det då nästan alltid enklare och snabbare att använda böcker enligt dem.

Sammanfattningsvis skulle jag vilja säga att alla studenterna anser att det finns tydliga problem med att använda webben i undervisningen, även om problemen varierar. Två elever uppger spontant och oberoende av varandra att det skulle behövas en bank med bra material, idéer, lektionsplaneringar m.m. speciellt för lärare och att det skulle avhjälpa många problem med webben om än inte alla.

Lärarna

ATT ARBETA SOM LÄRARE

Att det varierar mycket från skola till skola hur mycket lärare samarbetar styrks av de två lärare jag har pratat med. Den ena uppger att de jobbar en hel del i så kallade lärarlag och tar fram en hel del material gemensamt. Den andra läraren uppger att det inte finns några lärarlag i hennes skola just nu men att det kanske ändrar sig.

De två lärarna jag pratat med säger att de främst utgår från skolböcker eller egna böcker när de behöver tillgång till faktauppgifter. Pedagogiska idéer och inspiration kommer ofta från andra lärare som de jobbar med eller som de möter på studiedagar, annars från böcker. Eget material återanvänds ofta eller omarbetas något. Det material som de framställer i arbetet sparas i pärmar eller på datorn efter ämne och moment.

WEBBEN I LÄRARARBETET

Bara en av lärarna använder webben i arbetet. Hon använder webben till att slå upp aktuella bokrecensioner och artiklar om nya författare från de tidningsarkiv som skolorna har tillgång till. Det är aktuellt material hon är intresserad av. Litteraturhistoria ändras inte så mycket så böcker är för det mesta bättre än webben enligt hennes uppfattning. Den andra läraren säger att det bara finns en webbuppkopplad dator i hela skolan och det är inte värt besväret att använda den.

Lärarytbildarna

ATT ARBETA SOM LÄRARE

För två av tre lärarytbildare som jag talade med var det länge sedan de faktiskt arbetade som lärare, därför hade de inte så mycket synpunkter på det här området. Några hade de dock: Det varierar kraftigt hur mycket lärare samarbetar och delar på material. Vissa skolor jobbar med att ta fram mängder av material i lag, andra inte alls. Pedagogiska tips och idéer får man främst från andra lärare, på studiedagar och från läromedel

WEBBEN I LÄRARARBETET

Enligt två av utbildarna är det svårt att hitta bra material på Internet. Ofta är materialet man hittar av för dålig kvalitet eller på fel nivå, eller har "korkad" pedagogisk inriktning. En av dem säger att det skulle finnas ett behov av materialsamlingar på nätet, men att lärare är konservativa så det är svårt att införa nya arbetsmönster. Få skolor publicerar material elektroniskt, men det finns ett par.

Prototypen

Allmänt

Nedan följer synpunkter och reaktioner från försökspersonerna beträffande prototypen och dess användargränssnitt. Redogörelsen blir med nödvändighet ganska detaljerad. Jämför gärna synpunkterna med de olika versionerna av gränssnitten. Se avsnittet *Användargränssnittet* i kapitlet *Om prototypens uppbyggnad*. Bäst är förmodligen att jämföra med webb-versionerna [30].

Studenterna

SÖKGRÄNSSNITTET

Något som alla var överens om var att det bör finnas möjlighet att söka efter material på en viss nivå, till exempel material för mellanstadiet. Eftersom nivåbedömningar är en fråga om tycke och smak bör nivåindelningen vara ganska grov. Nästan alla tyckte också att det bör gå att söka på en viss typ av material eller ett visst avsett användningsområde.

I prototypen hade jag slagit ihop nyckelord och ämnesklassificering till ett fält för att få ett mindre antal fält att söka på. Flera försökspersoner tyckte dock att man borde dela upp det på två olika fält. Det ena fältnamnet bör då heta "ämne" och inte "ämnesklassificering".

Fältet *Header* är onödigt enligt tre studenter. De andra anser att det bör byta namn mot "dokumentinformation" eller "kataloginformation".

Fältet *hela texten* bör bytas ut mot "någonstans i texten" eftersom det inte stämmer med rubriken som är *skall finnas i*. En annan föreslagen variant är att byta rubriken *skall finnas i* mot "sök i".

Att kunna söka i *Regionen* behövs nog inte. Fälten bör heta "Egna skolan" i stället för *skolan* och *hela landet*.

Det bör finnas hjälptexter enligt flera testpersoner. Detta finns men jag ville inte ha med dem i försöket eftersom jag ville se hur mycket försökspersonerna förstår av gränssnittet utan "manual".

Något som alla var överens om var att det är bra att kunna göra sökningar på speciella fält. Sökfraser verkar också vara ett naturligt begrepp att jobba med för studenterna, även om det i ett fall ledde till att försökspersonen skrev in en lång sökfras och sedan tyckte att datorn borde ha hittat ett dokument som handlade om sökfrasen men som inte innehöll den exakt.

Knappen för att starta en sökning bör vara tydligare enligt flera testpersoner. Större och gärna med en avvikande färg.

Den första studenten som prövade prototypen tyckte att logiska operatörer mellan sökfrasfälten var onödigt krångligt. "Det blir för många val – gör det enklare!" Den andra versionen av gränssnittet utan logiska operatörer mellan sökfrasfälten verkade fungera bättre. En av försökspersonerna tyckte dock att det inte var helt klart hur fälten hängde ihop men det var inte svårt att förstå när någon berättade eller om det hade stått tydligare, enligt studenten. Till en ny version bör kopplingen mellan sökfrasfälten göras tydligare.

Enstaka försökspersoner har också haft ytterligare idéer. En testperson ansåg att det bör gå att söka vidare i en träffmängd för att förfina sökningen. Detta stöds inte av WAIS-protokollet men väl av Z39.50. Om det är möjligt att göra beror på vilket databasprogram man använder som grund för systemet. En annan testperson ansåg att det bör finnas en ämnesmeny att välja i först och att man sedan söker inom de kategorierna, som YAHOO. Detta är svårt att automatisera om man inte har fasta ämnesord som alla håller sig till. Om det går beror också på vilket databasprogram man använder. Ytterligare en försöksperson tyckte att det kanske skulle räcka med bara två sökfrasfält: "Jag skulle i alla fall aldrig använda fler". Frågan är

om andra skulle det. Detta är svårt att veta om inte gränssnittet prövas mycket mer med mer material inlagt i dokumentbasen.

TRÄFFPRESENTATIONEN

Alla studenterna var överens om att det var bra med en informativ och fyllig träffpresentation.

Två försökspersoner tyckte dock att det var viktigt att de var korta. "Korta presentationer ger mig bättre överblick" tyckte en av försökspersonerna. En annan försöksperson tyckte att det borde fungera som andra sökmotorer; där är ju presentationerna korta. Samma försöksperson föreslog att man skulle ha en övre gräns på kanske 4–5 rader. Den andra försökspersonen ville ha en mycket kort sammanfattning först med en länk till en fylligare sammanfattning och därifrån möjlighet att se hela dokumentet.

Avsett användningsområde eller typ av material bör framgå anser alla försökspersonerna. De vill kunna se om det är frågan om en laboration, en elevövning, ett lektionsförslag eller något annat.

Vilken nivå materialet är avsett för bör tydligt framgå enligt någon grov indelning. Någon bra indelning har försökspersonerna dock svårt att uppge. Kanske är 1–3, 4–6, 7–9 inte så dumt men indelningen av grundskolan i tre stadier är borttagen så det är en känslig fråga.

Språk bör tydligt framgå anser alla de blivande språklärarna.

En försöksperson vill kunna se när texten lades in i systemet. En annan vill kunna se längden på texten på något sätt.

Ett par försökspersoner anser att det bör finnas länkar till relaterat material, antingen i sammanfattningen eller i själva texten.

En försöksperson menar att det ibland kan vara svårt att ange ämne för material som är ämnesövergripande. En annan försöksperson säger att författare möjligen inte är så intressant.

ÖVRIGT

Alla försökspersonerna i den här kategorin anser att det bör gå att få utmatning både på webben och i ordbehandlingsformat. Alla anser att det är viktigt att kunna plocka delar och att bearbeta materialet. Då bör materialet vara i ordbehandlingsformat enligt dem.

Lärarna

SÖKGRÄNSSNITTET

Den första läraren som fick pröva prototypen fick pröva den första versionen av sökgränssnittet. Hon tyckte att booleska operatörer mellan sökfrasfälten var onödigt och krångligt. Hon tyckte att man lika gärna kunde göra flera sökningar om man ville uppnå samma funktion som logiskt eller.

Båda lärarna var överens om att det bör gå att ange nivå på materialet man söker.

Header kanske kan bytas ut mot "dokumentbeskrivning" enligt en av lärarna. Båda tyckte att namnet *header* var dåligt.

"Flera sökfrasfält gör det krångligare att lära sig men det är bra att man kan ha det.", sa en av lärarna.

Man bör skilja på ämne och nyckelord. Det bör heta ämne och inte ämnesklassificering.

TRÄFFPRESENTATIONEN

Båda lärarna tyckte att avsedd nivå på materialet bör finnas som en egen rubrik. Storlek i antal sidor vore också bra enligt båda. Båda lärarna anser också att ämne och avsett användningsområde bör ha egna rubriker.

Den ena läraren tycker att det är viktigt med en kort och kompakt sammanfattning, speciellt om man får många träffar, inte längre än 5–6 rader per träff.

ÖVRIGT

Båda lärarna anser att presentationen av träffar i huvudsak är bra. De vill båda ha mer informationsrika sammanfattningar än vad som är vanligt på webben.

Båda anser också att det är viktigt att kunna plocka bitar och använda. Utmatning bör därför definitivt finnas i ordbehandlingsformat.

Läroarbetsbildarna

SÖKGRÄNSSNITTET

Den ena av läroarbetsbildarna hade nästan aldrig sökt förut på webben och behövde få sökgränssnittet mycket noggrant förklarad. Fältbegreppet var en helt ny idé, liksom idén med flera inmatningsfält för sökfraser. Personen hade tidigare bara fyllt i ett antal sökord i någon vanlig sökmotor och visste inte ens om den använde "logisk och" eller "logisk eller" mellan orden. Tyvärr hade inte försökspersonen så mycket kommentarer om det nya gränssnittet, kanske för att personen inte ville verka dum. Åsikter om sökgränssnittet kommer nästan uteslutande från de två andra läroarbetsbildarna.

Båda de två andra läroarbetsbildarna tycker att alla fält som finns är bra utom "header". Den ena tycker att man skall ta bort det och ha "sammanfattning" i stället. Den andra tycker att man bör byta namnet från header till "dokumentinformation".

Den ena läroarbetsbildaren tycker inte att nivå och syfte bör ha egna fält; "det ska vara lite tuggmotstånd för lärarna". De två andra tycker att det vore bra men ingen vet hur man skall dela in nivån. I språk bör man dessutom kanske skilja på språklig nivå och på innehållslig nivå enligt en av lärarna.

Knappen för att starta sökning bör vara tydligare enligt två av försökspersonerna. Den bör dessutom ha kortkommando på "retur-tangenten".

Man bör ändra sökfältsrubriken *skall finnas i* till "Sök i" för att göra gränssnittet tydligare enligt en av försökspersonerna.

En av försökspersonerna anser att man bör skilja på ämne och nyckelord.

Region behövs nog inte enligt en av försökspersonerna.

En av försökspersonerna tycker dessutom att det borde finnas möjlighet till logiska eller-kopplingar mellan fält. Jag visade försökspersonen den första versionen av sökgränssnittet och det sökgränssnittet tyckte försökspersonen var bättre.

TRÄFFPRESENTATIONEN

Två av försökspersonerna tyckte att det bör finnas möjlighet till en lista med relaterat material via en knapp i träffpresentationen. "More like this" som i sökmotorn Excite vore jättebra om det fungerade.

En av lärarutbildarna tycker inte att avsedd nivå och avsett användningsområde skall ha egna rubriker. Den informationen kan möjligen stå i löpande text. De andra två försökspersonerna tycker att avsedd nivå och avsett användningsområde skall ha egna rubriker.

En av försökspersonerna tycker också att det bör finnas länkar till recensioner av materialet och länkar till diskussionsunderlag.

En av försökspersonerna tycker det vore bra om man kunde få se viktiga fraser – "nyckelfraser" ur texten.

ÖVRIGT

Alla tre försökspersonerna i den här kategorin tycker att det är bra med en fyllig träffpresentation. Det är också bra att man kan söka på speciella fält.

Övriga Synpunkter

Blivande lärare

OM SYSTEMET

Alla studenterna anser att det finns ett behov; systemet skulle vara bra att ha.

En del av dem tycker att det bör finnas en hierarkisk ämneskatalog, som YAHOO, till systemet som komplement till det sökgränssnitt de har fått pröva.

Flera av studenterna tror att det kan vara problem att få folk att lägga upp material. "Vem har tid med det?", undrar en blivande lärare. Flera studenter tror också att det kan vara problem med att få lärare att dela med sig av sitt material, liksom det kan finnas tekniska problem med själva publiceringen som ställer till problem.

Upphovsrättsfrågor måste redas ut ordentligt innan ett sådant här system kan tas i bruk anser två studenter. Detta är en fråga som dock gäller webben i allmänhet och det ligger långt utanför ramen för det här examensarbetet att försöka besvara sådana frågor.

En av studenterna tror att det kan bli svårt att komma överens om nivåklassificering, nyckelord etc. Det kan också bli problem med att folk lägger in "dålig" information.

"Vore bra om det främjade kontakter på något vis..." tycker en student.

Någonting som alla är överens om är att det är viktigt att materialet är plattformsoberoende, att man kan omarbete materialet och att man kan plocka delar. Att innehållet eller strukturen i dokumenten är detaljerat uppmärkt är ingenting någon fäster större vikt vid. Bara man kan få in dokumentet i sin favoritordbehandlare är det bra.

OM MATERIALET

Alla studenterna är överens om att det stora behovet är av lärarrelaterat material som lektionstips, övningsmaterial, idéer för hur man kan lära ut ett visst moment och liknande. Flera elever anser att faktamaterial redan finns i tillräcklig mängd på andra ställen som är enklare och snabbare att hitta än Internet t.ex. böcker.

Förslag på material som vore bra att ha är: övningsmaterial, lektionsförslag, texter och övningar för elevbruk, aktuellt material som dagstidningar, lokala betygskriterier från hela landet, lokala arbetsplaner från hela landet. Det bör kunna finnas kommersiellt eller sponsrat material inlagt men utmärkt med distributör.

Lärare

OM SYSTEMET

Båda lärarna anser att det finns ett behov av systemet. En lärare säger: "Det skulle kunna vara mycket användbart med en stor bas med material av lärare för lärare".

En lärare säger att man inte vill att elever skall ha tillgång till materialet hur som helst. Det är heller inte säkert att allt material en skola lägger upp skall vara tillgängligt utanför skolan.

En lärare tror också att det är svårt att göra material som skall läggas ut på Internet om man inte har fått någon utbildning i hur man gör. Det kan vara ett problem.

OM MATERIALET

Båda lärarna anser att man bör koncentrera sig på lärarproducerat material, som lektionsidéer, laborationer, grupparbeten och liknande. Rent faktamaterial finner man lämpligen på andra ställen. En lärare menar dock att det kan variera mellan ämnen. Litteraturhistoria ändras inte så mycket över en tioårsperiod medan andra ämnen som samhällskunskap kräver ständig tillgång till aktuella fakta.

Det är viktigt att man kan redigera materialet som man vill och att man kan plocka delar, det är båda lärarna överens om. En lärare tycker att det är viktigt att information som man söker på är uppmärkt men att det andra inte spelar så stor roll, "bara man kan få in det i Word".

Läroinnehåll

OM SYSTEMET

Behovet av ett sådant här system finns. Enligt en av läroinnehållarna behövs både mer lärarrelaterat material och bättre söksystem.

Två av läroinnehållarna tycker att det även borde finnas en ämneskatalog som komplement till prototypens söksystem.

Ett problem som en av läroinnehållarna kan tänka sig är att lärare inte är vana att skriva och publicera. Dessutom är de ett konservativt släkte. Man behöver stöd både från skolläda och politiker om ett sådant här system skall kunna få genomslag.

En av utbildarna tycker också det är viktigt hur man löser upphovsrättsfrågor. Det finns många potentiella problem där enligt denne.

OM MATERIALET

En av lärarutbildarna säger: "Bra material vore: läromedel, pedagogiska nyheter som tips om: nytt material, ny litteratur, information om pedagogiska forum, fortbildning och forskning. Samlade lokala arbetsplaner vore bra. Det skulle också behövas ett diskussionsforum för kursplaner och lokala arbetsplaner."

Två av lärarutbildarna säger att de skulle vilja ha tillgång till mer material relaterat till pedagogisk forskning.

SLUTSATSER OCH KONSEKVENSER FÖR DET FÖRESLAGNA SYSTEMET

Om lärares behov i stort

Det tycks stå helt klart att lärare har behov av kvalificerat material speciellt skrivet för lärare. Det största behovet verkar vara av läromedel, lektionsidéer, elevövningar m.m. Rent faktamaterial hittar de bättre på andra ställen än via dator enligt en stor del av de personer som jag har varit i kontakt med.

De flesta av försökspersonerna har haft ganska liten vana av att använda webben och har därför ett behov av sökverktyg som är lätta att förstå för personer som sällan eller aldrig har gjort sökningar i dokumentbaser tidigare. Försökspersonerna är också beroende av att sökträffar presenteras på ett effektivt sätt så att det går att sälla träffar snabbt och med noggrannhet.

Ett annat stort behov är att kunna ha plattformsoberoende material. Lärare har behov av att inte bara kunna använda färdigt material utan av att plocka delar och omarbeta material. Detta vill de kunna göra i sin favoritordbehandlare. Utmatning från systemet i exempelvis RTF-format är därför nödvändigt. Det verkar också som om de flesta anser att det här är den viktigaste möjligheten.

Om prototypens sökgränssnitt

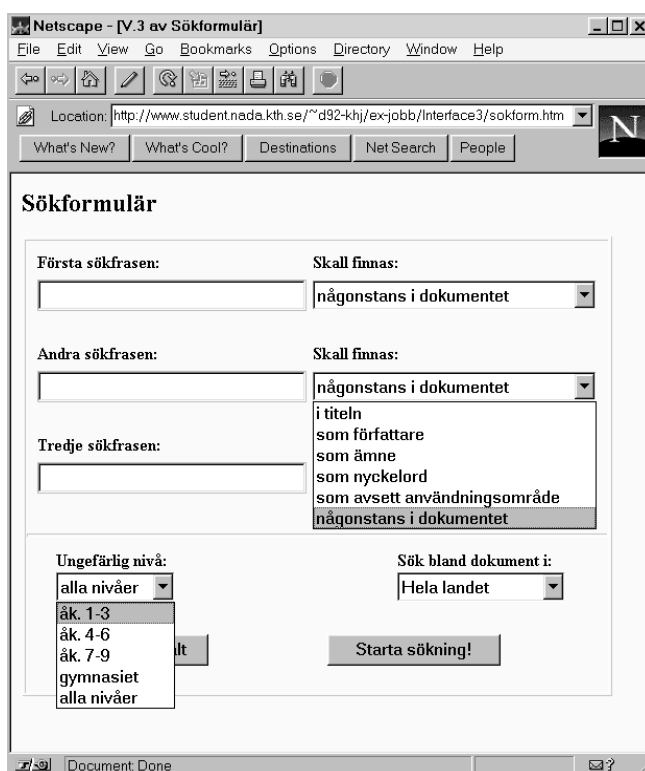
I stort sett upplevdes sökformuläret positivt. De två viktiga begreppen sökfraser och sökfält förstod alla försökspersonerna efter lite förklaring och tyckte de var bra begrepp att arbeta med. Bara en testperson har saknat den första versionens möjligheter att ställa in vilken logisk koppling det skall vara mellan sökfräsfälten så jag håller nog fast vid att den möjligheten mycket sällan behövs och därför inte bör vara kvar. Dock kan man ju tänka sig att ha flera olika gränssnitt med olika många funktioner i de olika varianterna.

Att kunna ange avsedd nivå på materialet var något som nästan alla försökspersonerna ville ha möjlighet till. Denna funktion bör definitivt läggas till i gränssnittet. Förmodligen bör den ligga utanför sökfraserna som en kombinationsruta eftersom den här funktionen förmodligen kommer att användas flitigt.

Att kunna se det avsedda användningsområdet för materialet var också något som alla försökspersonerna ansåg viktigt. Antalet potentiella användningsområden som olika material kan ha kan dock bli mycket stort varför det kan bli svårt att standardisera. Om det skall gå att söka på avsett användningsområde måste det bli frågan om att ange en fritextsträng vid sökning. Förslagsvis gör man ett nytt fält att söka i databasen som heter avsett ämnesområde.

Rubriken över menyrutan för sökfält bör bytas ut. Mitt förslag är att man byter *skall finnas i* till *skall finnas:* och att man sedan ändrar fältnamnen så att de passar ihop med den nya rubriken.

Rubrik och fältnamn bör ses över en gång till. Nya rubriker i träffpresentationen bör bli avsedd nivå, ämne, avsett användningsområde. Hur gränssnittet ändrades i sin tredje version finns beskrivet i avsnittet Användargränssnittet version 3 i kapitlet Om prototypens uppbyggnad. Se också Figur 7.



Figur 7. Den tredje versionen av sökförmuläret.

Lärares behov kontra gratisprogram och standarder

Det är helt klart att lärare har speciella behov av olika slag. Problem uppstår naturligtvis då dessa behov inte stöds på något standardiserat sätt i de fritt tillgängliga program som finns. Exempel på ett sådant behov som har kommit fram under användarstudierna är möjligheten att få se en ämneskatalog över dokumentbasen och att kunna se vilken nivå materialet är avsett för. Eftersom det inte finns någon befintlig standard för hur det här bör göras bör man vara försiktig med att ta med dessa funktioner.

Om en sådan funktion trots det anses så viktig att den bör tas med måste man fastslå hur den skall implementeras. Om det handlar om att ha med speciella metadata i alla dokument måste man komma överens om hur de skall märkas upp. Denna överenskommelse måste sedan följas om systemet skall fungera. Detta gör att det blir mer att hålla reda på för den som vill publicera dokument i systemet.

Andra önskemål eller behov går inte att lösa alls med dagens verktyg t.ex. att ange att man vill ha material som liknar ett visst dokument. Då kan man bara vänta och hoppas att framtidens verktyg löser problemet.

BILAGA 1

UTVÄRDERING AV PROTOTYP OCH INTERVJU

INTERVJU

Datorer

Hur stor tidigare datorvana har personen?

- ◆ Till vad används datorn främst?
- ◆ Egen dator?
- ◆ WWW-vana?
- ◆ Till vad används WWW främst?
- ◆ Vana av sökning på WWW?
- ◆ Hur stor del av tiden går åt till sökning?
- ◆ Erfarenheter av sökning
- ◆ Hur används datorn i skolarbete i dag?
- ◆ Används WWW till själva lärarbetet?
- ◆ Vad saknar personen på WWW idag?

Skolarbete

Vilken typ av arbetsmaterial används främst (eget, skolböcker, sådant som elever själv hittar)

- ◆ Var hittar du främst faktamaterial till lektioner eller kursavsnitt? (egna böcker, läroboken, internet osv)
- ◆ Var hittar du främst pedagogisk inspiration/ information? (andra lärare, artiklar, egna erfarenheter)
- ◆ I vilken mån delar lärare på material de själva har framställt?
- ◆ I vilken mån jobbar lärare tillsammans med att framställa material?
- ◆ Hur hanteras allt det material som framställs?

UTVÄRDERING AV PROTOTYPEN

Uppgift-fysik

Plocka fram en text som handlar om ljus och om ljusets hastighet. Om en sökning ger mer än en träff, skall sökningen förfinas tills bara det önskade dokumentet återstår.

Pröva gärna sökningar med flera sökfraser och pröva att använda fältsökning.

Uppgift engelska

Hitta en lämplig undervisningstext. Du har hört att det skall finnas en version av Aesopos fabler som är preparerad med svenska gloslistor. Om en sökning ger mer än en träff, skall sökningen förfinas tills bara det önskade dokumentet återstår.

Pröva gärna sökningar med flera sökfraser och pröva att använda fältsökning.

Sökformuläret

Pröva fältsökning – förfining av sökfråga

Logiska operatorer – förfining av sökfråga

Byta sökregion – inga träffar i fel område

Töm fält – vad gör den? Vad borde den heta?

Resultatsidan

- ◆ Allmänna frågor om hur sökresultat bör presenteras med utgångspunkt från prototypen. Vilken information bör finnas? Hur fyllig skall den vara?
- ◆ Vad vill man se i en resumé? Vad är det viktigaste att den innehåller?

Märkningen / dokumenten

Vilken metadata bör finnas?

Fysik

Vilken typ av material skulle du vilja ha tillgång till? Populärvetenskapligt?

”riktig fysik”? Labbar? Lektionsupplägg? Läromedel?

Hur viktigt är det med formelmärkning?

Hur skulle du vilja att formler fungerade?

Hur viktigt är det att kunna plocka delar och använda?

Engelska

Vilken typ av material saknar du mest? Skulle vilja ha mer av?

Lektionsupplägg? Övningar? Texter?

Hur viktigt är det att kunna märka upp grammatik?

Hur viktigt är det att kunna plocka delar?

Hela det tänkta systemet

Spontana synpunkter?

Funderingar kring dess funktion?

Hur skulle du tänka dig att administrationen skulle fungera?

LITTERATURFÖRTECKNING

INLEDNING

1. D. BROADY (1992): *Kunskapsverkstaden – Lokala dokumentbaser som arbetsverktyg för lärare*, Datorn i utbildningen, nr. 2, 1992. Innehåller ett antal viktiga synpunkter på vad ett datorstöd för skolbruk bör ha för egenskaper.
2. M. ANDERSSON (1997): *Internet som läromedel – ett alternativ till boken?* Ett examensarbete utfört inom utbildningsprogrammet "Praktisk pedagogisk utbildning 40 p" vid högskolan i Växjö.
3. *D-lib program – research in digital libraries*,
<http://www.dlib.org/dlib.html>. Verifierad 1998-02-15.

OM ANVÄNDARGRÄNSSNITT OCH SÖKNING

4. S. D. PAYETTE, O. Y. RIEGER (1997): *Z39.50 – The users perspective*, D-Lib Magazine, april 1997. En artikel om en användarstudie bland studenter och forskare om möjligheten till att göra parallella sökningar i olika databaser. Undersökningen strävar efter att komma fram till ett bättre gränssnitt för sökning än de man har.
5. A. HOCKLEY, POLLOCK (1997): *What's wrong with Internet searching*, D-lib Magazine, mars 1997. Artikelförfattarna utgår ifrån en användarstudie av personer med mycket liten webb- och Internetvana för att visa på de brister i gränssnittet som dagens stora webbsökmotorer uppvisar. Artikeln ger också exempel på olika vanliga föreställningar om vad Internet egentligen är för något.
6. B. SHNEIDERMAN, D. BYRD, W. B. CROFT (1997): *Clarifying search*, D-lib Magazine, januari 1997. Artikeln tar upp många av de brister som gränssnitten i dagens webbsökmotorer uppvisar: Bristande tydlighet i gränssnittet, bristande konsekvens, svårt att skapa en bra mental modell m.m. Sökprocessen behandlas också och delas in i fyra steg. I artikeln ges konkreta förslag på vad man bör tänka på när man gör gränssnitt till söksystem.

TEKNISK UPPBYGGNAD AV SÖKFUNKTION

7. H. CHU, M. ROSENTHAL (1996): *Search A comparative study nad evaluation Methodology*, ASIS 1996 Annual conference proceedings, oktober 19-24 1996. En artikel som jämför tre vanliga sökmotorer på webben samt föreslår en metod att utvärdera sökmotorer på basis av deras undersökning.
8. S. HAMMER, J. FAVARO, I. SISTEMI (1996): *Z39.50 and the world wide web*, D-Lib Magazine, mars 1996. Beskriver hur Z39.50 kan användas i hop med webben för att få möjlighet till sökning av hela nätet, från olika källor, från ett och samma användargränssnitt. Även en del om möjligheten att söka samma databaser med olika användargränssnitt, anpassade efter olika personers behov.
9. C. A. LYNCH (1997): *The Z39.50 information retrieval standard – part I: a strtegic view of its past, present, and future*, D-Lib Magazine, april 1997. En historisk genomgång av Z39.50, hur den har utvecklats, vad som har styrt utvecklingen och var den är på väg. En bra översikts och bakgrundsartikel.
10. J. A. KUNZE, R. P. C. RODGERS: *Z39.50 in a nutshell*, <http://www.lhncbc.nlm.nih.gov/csb/pubs/rodgers/z39.50/z39.50.html>. Verifierad 1998-02-15. Teknisk översikt över Z39.50-protokollet.
11. *CNIDIR Isite*, <http://vinca.cnidr.org/software/Isite/Isite.html>. Verifierad 1998-02-15.
12. *Free WAIS-SF – The enhanced free WAIS distribution*, <http://ls6-www.informatik.unidortmund.de/ir/projects/freeWAIS-sf/index.html>. Verifierad 1998-02-15.

OM METADATA

13. S. WEIBEL: *Metadata: the foundation of resource description*, D-lib magazine, juli 1995. Artikeln tar kortfattat upp vad metadata är, vad man använder det till samt vilka olika standardiseringsprojekt som pågår.
14. S. WEIBEL, R. IANNELLA (1997): *The 4th Dublin core metadata workshop report*, D-Lib Magazine, juni 1997. Rapport från det pågående arbetet med the Dublin core. Ger en viss överblick över problemen med en global standard för metadata för dokument. Artikeln ger också en inblick i vilka olika strategier man kan anta vad gäller exempelvis komplexitet.

OM LAGRING OCH LAGRINGSFORMAT

15. TEI P3 [Kap.2]: *A gentle introduction to SGML, Chicago/Oxford: Text Encoding Initiative*, URL: <http://etext.lib.virginia.edu/TEI.html>. Verifierad 1998-02-15. Precis vad titeln antyder. Förmodligen en bra ställe att starta på om man inte vet så mycket om SGML.
16. D. BROADY (1996): *Digitala utgåvor av August Strindbergs verk – Inledande överväganden*, Version 3, Stockholm, KTH, NADA/CID. Mycket grundlig redogörelse för de överväganden som behövs i samband med ett projekt för att lägga upp August Strindberg i digital form. Mycket användbar introduktion för att se vilka överväganden som behöver göras när man sätter upp en digital dokumentbas.
17. D. BROADY, H. HAITTO (1996): *Intenet and the humanities: the promises of integrated open hypermedia*, Stockholm, KTH, NADA, IPLab-106. Allmänt om SGML, DSSSL, HyTime, TEI m.fl.
18. E. VAN HERWIJNEN (1994): *Practical SGML Second Edition*, London Kluwer Academic Publishers. Bra bok om SGML som tar upp grunderna och de flesta funktionerna i SGML.
19. <http://www.jclark.com/dsssl/>. Verifierad 1998-02-15.
20. *CPAN Multiplex Dispatcher*, <http://www.perl.com/CPAN>. Verifierad 1998-02-15.
21. *Cost*, <http://www.art.com/cost/>. Verifierad 1998-02-15.
22. *The whirlwind guide to SGML tools and vendors*, <http://www.falch.no/people/pepper/sgmltool/convert.htm>. Verifierad 1998-01-20.

UTMATNING OCH KONVERTERING

23. J. PRICE-WILKIN (1997): *Just-in-time conversion, just-in-case collections*, D-lib Magazine, maj 1997. Artikeln behandlar de avväganden som bör göras när man väljer mellan att konvertera dokument till utmatningsformat på begäran eller en gång för alla vid lagringen. Artikelförfattaren rekommenderar konvertering på begäran vid stora dokumentmängder och till format som ändrar sig som t.ex. HTML.
24. *Länkskafferiet*, <http://munin.ub2.lu.se/skolverket/>. Verifierad 1998-02-15.
25. *TEI Application Page (by Type of Project)*, <http://www-tei.uic.edu/orgs/tei/app/topics.html>. Verifierad 1998-02-15.

TEI OCH TEI-LITE

26. *TEI P3, Guidelines for Electronic Encoding and Interchange*, Chicago/Oxford: Text Encoding Initiative, URL: <http://etext.lib.virginia.edu/TEI.html>. Verifierad 1998-02-15. Ett standarddokument som faktiskt är mycket läsbart! Innehåller en hel del förklarande text och innehåller naturligtvis standarden i sin helhet samt utdrag ur DTD:n.
27. *Text Encoding Initiative homepage* URL: <http://www.uic.edu/orgs/tei/>. Verifierad 1998-02-15. Här finns länkar till allt möjligt TEI-relaterat samt officiell TEI-information.
28. LOU BURNARD, C. M. SPERBERG-MCQUEEN: *TEI Lite: An Introduction to Text Encoding for Interchange*, URL: <http://www-tei.uic.edu/orgs/tei/intros/teiu5.html>. Verifierad 1998-02-15. Bra ställe att lära sig TEI-Lite på.
29. D. SEAMAN (1997): *The user community as responsibility and resource – building a sustainable digital library*, D-Lib Magazine, juli 1997. Artikeln argumenterar för att utbildning av användare, samt täta kontakter med användarna ger ovärderlig återkoppling på det man gör. Återkoppling från användare möjliggör att man kan utveckla systemet åt rätt håll samt att användarna känner sig delaktiga när de märker att de kan påverka.
30. <http://www.student.nada.kth.se/~d92-khj/ex-jobb/toproto.htm>. Verifierad 1998-02-15.

OM PROGRAMMERINGSTEKNISKA DETALJER

31. *WWW-to-PAT gateway: Exploiting an SGML-aware system through the web*, <http://sansfoy.hti.umich.edu/www-to-pat> Verifierad 1998-02-15. En steg för stegbeskrivning av hur man har gjort ett webbgränssnitt till en PAT-databas. Artikeln innehåller mycket detaljerad information om hur man har gått till väga. PAT är Open Texts databas/söksystem och är inte PD.
32. *sgrep – search a file for a structured pattern*, <http://www.cs.helsinki.fi/~jjaakkol/sgrepman.html>. Verifierad 1998-02-15.
33. W3C-group: *Mathematical Markup Language – W3C Working Draft*, URL: <http://www.w3.org/pub/WWW/TR/WD-math/>. Verifierad 1998-02-15. Innehåller en hel del diskussion om problemet med märkningsformat för matematik samt ett förslag till XML-DTD för matematik som nog måste anses vara state of the art. Förslag är bara ett utkast än så länge i väntan på att W3C-working group skall anta den som en rekommenderad standard. Ett dokument som ger bra insikt i problematiken.

34. N. F. A. M. POPPELIER, E. VAN HERVIJNEN, C. A. ROWLEY (1992): *Standard DTDs and Scientific publishing*, EPSIG News, september 1992. Argumenterar för en uppsättning standard-DTD:er för vetenskaplig publicering, samt diskuterar problemet med märkningsformat för matematik. En artikel som ofta refereras i sammanhanget.