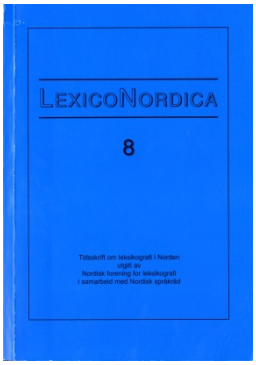
LexicoNordica

Titel:	eXtensible Markup Language (XML) og leksikografi
Forfatter:	Franziskus Geeb
Kilde:	LexicoNordica 8, 2001, s. 167-183
URL:	http://ojs.statsbiblioteket.dk/index.php/lexn/issue/archive



Betingelser for brug af denne artikel
Denne artikel er omfattet af ophavsretsloven, og der må citeres fra den. Følgende betingelser skal dog være opfyldt:

- Citatet skal være i overensstemmelse med „god skik“
- Der må kun citeres „i det omfang, som betinges af formålet“
- Ophavsmanden til teksten skal krediteres, og kilden skal angives, jf. ovenstående bibliografiske oplysninger.

Søgbarhed
Artiklerne i de ældre LexicoNordica (1-16) er skannet og OCR-behandlet. OCR står for ’optical character recognition’ og kan ved tegngenkendelse konvertere et billede til tekst. Dermed kan man søge i teksten. Imidlertid kan der opstå fejl i tegngenkendelsen, og når man søger på fx navne, skal man være forberedt på at søgningen ikke er 100 % pålidelig.

Franziskus Geeb

eXtensible Markup Language (XML) og leksikografi

1. XML

XML (eXtensible Markup Language) er en standard fra W3C (www.w3.org), som står for alle internetstandarder som f.eks. HTML. XML foreligger i en godkendt version og er dermed en standard, som kan bruges uden indskrænkninger. Direkte til XML er der knyttet en hel del yderligere standarder, som enten er afledt fra XML eller udviklet samtidig til brugen af XML (XSLT, XSL-FO, XPath, XLink, XPointer, XQL, ...). Dertil kommer en større del af tredjepartsprodukter, som enten tilbyder software til bearbejdning af XML (editor etc.) eller leverer produkter til visning eller viderebehandling af XML-dokumenter.

XML er en meget åben standard for definitionen af markupsprog. Markupsprog er ikke ægte programmeringssprog men derimod et redskab for markering af tekstdele til maskinel brug. Den tungeste version er kendt som SGML, den letteste og nu mest udbredte version af markupsprog er HTML. Teksten bliver delt op i enheder, som forsynes med mærkater ("tags"). Mærkaterne bliver kendetegnet med specielle tegn. Når denne tekst inkl. dens mærkater bearbejdes og f.eks. vises med en dertil beregnet software (f.eks. browser), vises som regel kun teksten. Mærkaterne bliver sat om til f.eks. layout så som fed tekst, overskrift, understreget tekst etc. Analogien fra et markupsprog til en datastruktur er åbenlyst. Ligheder og berøringspunkter mellem datastruktur og strukturerede data i leksikografiske opslagsværk (Geeb 1998:35) er lige så klare. Det er dermed et lille skridt fra leksikografiske data til et dertil egnet markupsprog.

HTML er den kendte version af et markupsprog, men det er langt fra det eneste markupsprog, som bruges! Det særlige kendetegn ved HTML er blandingen af mærkater med forskellig kvalitet. Mærkater som <b> for "fed tekst" er en layoutoplysning og ikke en tekststrukturel oplysning. De dertil beregnede mærkater som f.eks. for fremhævet tekst svarer mere til en tekststrukturel (baseret på indholdets struktur) oplysning. Dog bruges disse i dag næppe og HTML er dermed stort set blevet til et layoutsprog. Desuden – og det er HTMLs andet væsentlige kendetegn – er alle tags foruddefineret. Der er ikke mulighed for at udvide sproget med egne tags – f.eks. leksikografiske – , hvis standarden

og derudover langt størsteparten af den til produktion og brug af teksterne tilgængelige software skal følges. HTML er altså en nøje afgrænset mængde af layoutmærkater med lidt eller hhv. ingen tekststrukturel informationsværdi.

Dermed ville det være svært, at udtrykke følgende artikel fra et leksikografisk opslagsværk i HTML:

> **in.voice** (in´vois), *n.*, *v.*, **-voiced, voicing. --**n. **1.** an itemization of goods purchased or services profided, together with the charges and terms **2.** ...

Selvfølgelig kunne man bruge følgende kode:

```
<P ALIGN=LEFT STYLE="margin-left: 2cm; margin-bottom: 0cm"><B><FONT FACE="Times New Roman"><FONT SIZE=4>in.voice</FONT></B><FONT FACE="Times New Roman" SIZE="3"> (in&acute;vois), <I>n., v.</I>, <B>-voiced, voicing.</B> <B>--</B>n.<B>1.</B> an itemaization of goods purchased or services profided,togehter with the charges and terms <B>2.</B> ...</P>
```

Dog gengiver disse mærkater kun layoutet, de indeholder ikke udsagn omkring strukturen og indholdet af den leksikografiske tekst. Set fra informationsdesignet af den leksikografiske tekst er det dog underordnet, om lemmaet er fed eller kursiv eller ingen af delene. Det er meget mere relevant for leksikografen, at lemmaet er et lemma, altså indgangen til den i mikrostrukturen lagrede information.

Med CSS (Cascading Style Sheets) har man i HTML prøvet at adskille layoutoplysninger fra informationsstrukturen. Layoutet (fed, kursiv etc.) ligger her i særskilte filer, men stadigvæk har man til informationsstruktureringen kun de muligheder til rådighed, som HTML byder med de foruddefinierede og begrænsede tags. Tags til leksikografisk brug mangler helt. XML løser netop dette problem. I XML findes ingen foruddefinierede tags. Alle tags defineres af brugeren selv samtidig med at den leksikografiske tekst opdeles i informationsenheder. Disse tags har ingen layoutvirkning. Set med en standardsoftware som f.eks. Microsoft Internetexplorer >= 5.0 eller Netscape Navigator >= 6.0 ville man derfor ved et XML-dokument kun se mærkaterne og deres indhold og ellers ingen layout. Det er også grunden til en stor skuffelse hos mange brugere af HTML. XML er ikke en ypperlig udgave af HTML. Denne forhåbning bliver heller ikke indfriet af XHTML, den XML-baserede, meget strukturerede version af HTML. XML byder ikke på alle de layoutbegreber, som man mangler i HTML. Tværtimod er XML befriet af al layout, og informationsdesigneren (f.eks. en leksikograf) kan nu uforstyrret

arbejde med indholdet og dens struktur. – Layoutet af XML-filer kan skabes gennem f.eks. CSS-stylesheets eller gennem en senere transformation til HTML eller Adobes meget læservenlige format PDF (Portable Document Format). Men alt dette er inholdet af teksterne uvedkommende. Layoutet adskilles i et særligt trin, og XML-filen opfattes nu kun som en enhed til lagring af data. Det er derfor ikke uden grund, at XML-filer sammenlignes på visse områder med tabeller i databaser eller endda databaser.

En korrekt XML-fil (på elementniveau; der mangler en XML-indledning) er: <greeting>Hello World</greeting>

Men lige så korrekt er : <hilsen>Hello World</hilsen>

Indholdet af de to filer er ens, endda strukturen er ens, kun benævnelsen af informationsenhederne er forskellig. Følgende fil har en anderledes struktur, men i den konkrete produktion af information, som vises til brugeren, kunne den gengives på samme måde som de to første eksempler:

```
<greeting>
  <first>Hello </first>
  <second>World</second>
</greeting>
```

En XML-fil skal i det væsentlige kun opfylde følgende krav for at være en lovlig XML-fil (=wellformed XML):

1. Alle tags, som åbnes, skal også lukkes (eller være tom)
2. I mærkaterne (=elementernes navne) er der forskel mellem store og små bogstaver
3. Mærkaterne kan bygges i en hierarki af vilkårlig dybde og bredde, men mærkaterne skal lukkes i den rækkefølge, som de er blevet åbnet
4. Der må kun være et root-element (lige som <html> i HTML-filer)

XML-filer er som basis til datalagring meget åbne. To filer med samme indhold behøver derfor ikke at følge samme struktur. – XML-filer kan vises med hvilken som helst software, bare den kan bearbejde XML-filer! Dog ligger her problemet, da XML-Software som regel bliver lavet til enkelte opgaver på grundlag af et nogle komponenter. Disse kompo-

nenter findes i flere udgaver og er som regel gratis (Open Source). Dog skal man selv knytte dem sammen og tilrette dem efter opgaven. Leksikografiske XML-værktøjer findes endnu ikke. Derfor er leksikografen, som vil benytte sig af denne teknologi, nødt til at samarbejde med en tilsvarende udvikler eller selv at udvikle den fornødne software. Som alternativ kan man bruge standardsoftware, som allerede findes. Editorer kan producere og validere XML-filer, de helt nye internetbrowsere kan vise dem. Dog kan editionerne og browserne ikke benytte sig af den fulde styrke af XML. Den fremkommer ved brugen af de forskellige lag i den lagdelte XML-procesmodel (se afsnit 2 i denne artikel).

En væsentlig del af denne XML-procesmodel, som ligeledes gælder for leksikografiske data i XML, er muligheden for at transformere dataene. I denne transformationsproces kan data udvælges og derefter bearbejdes, tilføjes til nye dokumenter, slettes eller bare gives videre f.eks. til brugeren i et grafisk program. Transformationssproget, som er et programmeringssprog skrevet i XML er XSL, eXtensible Stylesheet Language og her nærmere betegnet XSLT, eXtensible Stylesheet Language Transformations. Dette sprog må ikke forveksles med CSS, Cascading Style Sheets, fra HTML. CSS er beregnet til layoutspecifikationer hvorimod XSLT er beregnet til processering af data med henblik på at udvælge og muligvis forandre disse data. – CSS kan meget vel bruges til at vise XML-filer, men det vil svare til en direkte visning af hele datafilens indhold uden videre bearbejdning. Netop i leksikografien og med de forskellige brugere og deres specifikationer som målgruppe er det derimod nødvendigt, at vise kun dele af hele datafilen. Ligeledes vil det i de fleste tilfælde være nødvendigt at vise de elementer af XML-filen, som udvælges, f.eks. i en speciel orden for at kunne skabe den ønskede mikrostruktur. CSS kan her bruges som layoutsprog, hvis dataene fra XML-filen skal omdannes ved hjælp af XSLT til HTML. Disse HTML-filer kan så videresendes f.eks. gennem internettet til brugeren, og her er CSS et udmærket valg til en centraliseret styring af layoutet. Fordelen med XSLT som værktøj til retrieval af de netop i det øjeblik fra brugeren søgte data fra en leksikografisk databasis i en XML-fil er, at man kan bruge denne teknologi på flere niveauer. Kay (2000:533) konkluderer her følgende fire trin i XSLT-stylesheets i forhold til deres opgave og virkemåde: "Fill-in-the-blanks stylesheets, Navigational Stylesheets, Rule-based stylesheets, Computational stylesheets". XML-baserede leksikografiske data kan dermed gengives på en nem måde (Fill-in-the-blanks), som minder om almindelig kendt HTML-markup, eller de kan bearbejdes og udvælges med en meget moderne funktionel programmering (computational).

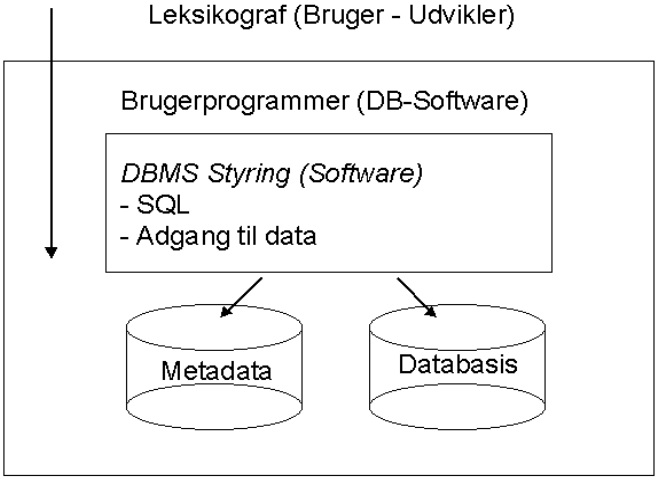
XML tilbyder ved siden af et meget struktureret og objektorienteret datagrundlag samt et tilhørende programmeringssprog til udvælgelse og bearbejdning af data on the fly, statisk eller dynamisk også et overordnet struktureringselement. Antallet af mulige tags og dermed af mulige elementer er i XML principielt ubegrænset. For leksikografiens vedkommende vil de dog selv med den mest avancerede makro- og mikrostruktur samt "repräsentatives" og "ergänzendes Lexemwissen" (Geeb 1997) kunne begrænses til højst et par hundrede forskellige tags per dokument og dermed det samme antal elementer. Ud over at antallet af forskellige elementer i hvert dokument kan begrænses er det med hensyn netop til makro- og mikrostrukturen nødvendigt, at kunne sætte elementerne i forhold til hinanden samt forsyne dem i disse relationer med kvalitets- og kvantitetskendetegn. – Grammatiske oplysninger forekommer nu engang som del af mikrostrukturen og som regel efter lemmaet. Til dette formål har XML i lighed med SGML, som er den ældre og større men mindre kendte og brugte bror af eXtensible Markup Language, muligheden for at tilknytte hver XML-fil en strukturdefinition. Denne strukturdefinition (Document Type Definition = DTD) knyttes til alle XML-dokumenter med samme informationsstruktur, altså f.eks. til alle leksemer fra en lemmaliste (hvert leksem som en XML-fil). Hermed svarer en DTD på mange måder til en datamodel fra databasesystemer. Både XSLT og DTD er del af XML-verdenen, men derudover findes der et stadig stigende antal af XML-baserede og -relaterede sprog, som til dels også ville kunne bruges i leksikografien (f.eks. SMIL: Synchronized Multimedia Intregration Language)

2. XML og databaser i leksikografien

XML kan forstås som en form for struktureret lagring af data eller informationer. På denne måde kan XML meget vel sammenlignes med de kendte relationale (eller også andre) databaser. Der er dog væsentlige forskelle mellem lagring af information eller data i databaser eller i XML. Databaser (og det gælder i det mindste relationale databaser, som stort set er de databaser, som bruges i leksikografien) er i det almindelige sprogbrug ikke kun den fysiske lagring og strukturering af data. Derimod forstås ved databaser der også selve softwaren, som typisk tilbyder en mangfoldig tilgang til dataene. Et kendte eksempel er Microsofts databaseprodukt Access. Alt det grafiske, knapperne, menuerne og de mange assistenter til gavn for brugeren er kun en overbygning, som ikke har ret meget med den virkelige lagring af dataene at gøre. Dog er

det netop denne overbygning, som gør de grafiske WYSIWYG-produkter på databaseområdet meget interessant for brugerne. Med lidt fagkundskab kan man i løbet af kort tid udvikle mindre databaser, som svarer til en relational datamodel. Dette gælder efterhånden også for de større, mere profesionelle databaser som Oracle eller MS SQL-Server. Grafiske faciliteter, som disse produkter ikke tilbyder, bliver så alligevel tilgængelige gennem software som MS Access ved hjælp af den fælles databasegrænseflade ODBC.

Datalagring samt informationsstrukturering i den leksikografiske produktionsproces burde ikke være noget problem ved brug af relationale databaser. Arbejdsgangen vil her – ved siden af mange andre leksikografiske overvejelser være – at udvikle en datamodel og dermed visualisere de enkelte informationsenheder og relationen mellem dem. En efterfølgende omformning til relationsskema samt normalisering vil betyde en effektivisering af informationsstrukturen. Derefter er omformningen i en tabelstruktur overskuelig og uden yderligere anomalier. Denne problemløsning bliver i dag brugt i mange leksikografiske sammenhænge på forskellige niveauer – med eller uden modellering i en Entity Relationship Diagram og efterfølgende normalisering. Som enkeltbrugersystem eller system på et enkelt netværk implementeres disse løsninger som regel uden problem. I distribuerede systemer som f.eks. internettet, hvor flere forskellige leksikografer arbejder sammen i samme projekt, kræver en databaseløsning som regel eksperthjælp på IT-siden. Leksikografen vil dermed som regel opleve den leksikografiske database på følgende vis:



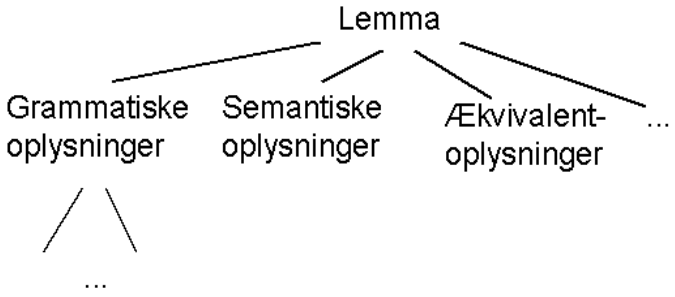
Leksikografen benytter sig som bruger eller udvikler nok mest brugerprogrammet (tit den grafiske brugergrænseflade). Denne sørger som re-

gel for den konkrete brug af de efterfølgende programmer, som giver fysisk adgang til dataene samt omformulerer databasesproget SQL (Structured Query Language – en ANSI standard) til denne fysiske adgang. Dette gælder såvel metadata (data med oplysninger omkring de leksikografiske data samt brugeroplysninger) som også den leksikografiske databasis.

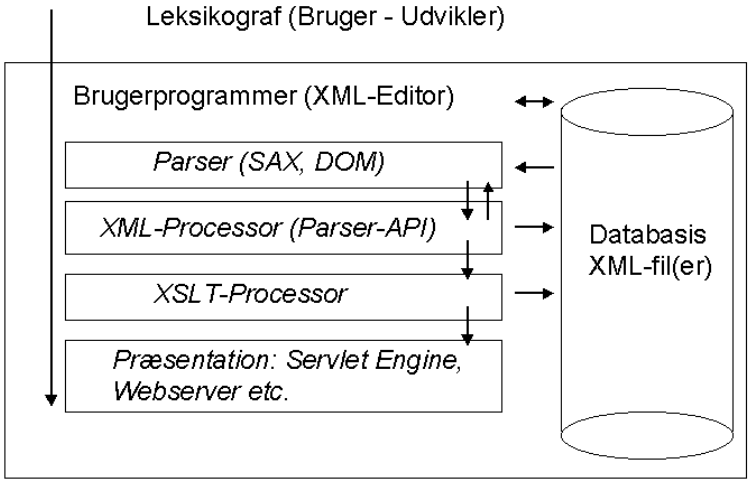
Eksport og import af data er her som regel afhængig af den konkrete software, men den rene databasis er i langt de fleste tilfælde ikke direkte tilgængelig for brugeren. Strukturdefinitionen (relationerne mellem informationsenhederne og dermed mellem tabellerne) ligger i semiprofessionelle og profesionelle systemer ligeledes i databasen, men de er igen kun tilgængelige gennem brugerprogrammer hhv. DBMS-styringen.

XML tilbyder her en helt ny tilgang til lagring og bearbejdning af leksikografiske data. Åbenlyst og dermed ikke del af den videre diskussion er XML som brug af transportmedium for data mellem applikationer. Det kunne være transport af leksikografiske data fra en database til en anden eller fra en database til et layoutprogram som f.eks. Adobe Framemaker, som tilbyder specielle XML-filtre. Hvor man tidligere arbejdede med kommaseparerede tekstfiler vil et sæt af XML-filer kunne erstatte denne opgave. XML-filerne er meget mere læsbare for brugerne samt mere åbenlyse i deres struktur. Desuden kan de direkte bearbejdes i en dertil egnet editor samt vises grafisk i forskellig slags software. Dermed er XML et optimalt transportmedium ikke mindst for leksikografiske data. Eksporten af data til XML fra en database er afhængig af dens brugerprogrammer, altså det, den tilbyder brugerne. I fremtiden vil der næppe kunne eksistere et databaseprodukt på markedet, som ikke tilbyder denne eksportfunktion til XML.

Af meget større betydning for leksikografien er dog den nye arbejdsproces ved dataproduktion, informationsstrukturering og lagring af data samt information ved en direkte brug af XML. Udgangspunktet er dog det samme. De leksikografiske informationer skal først omdannes til strukturerede data. Deres værdi som information tilkommer dem først efter en vellykket opbygning af denne struktur. Også ved XML gælder det om at skelne mellem de væsentlige enheder, som danner rammen omkring alle informationer (f.eks. lemma, ækvivalentoplylsning etc.). Dog opbygges ved design af XML-dokumenter ikke entiteter, som sættes i relation til andre entiteter, men enhederne sættes i en hierarkisk struktur. Udgangspunktet er dermed den enhed, som leksikografen vælger til at være udgangspunktet, f.eks. et lemma.



Strukturen og struktureringen er fri, men der vil altid kun være et rodelement. Toppen af hierarkiet er dermed altid defineret. Ny ved denne tankegang er for leksikografen, at ordbogsartiklen i den trykte form nu også kan udtrykkes i en informationsstruktur uden at de dertil fornødne data ligger i forskellige tabeller alt efter dataenes sammenhæng og hyppighed. Lemmaet og dets informationer kan dermed betragtes som et objekt. Bearbejdningen af det enkelte lemma bliver dermed mere overskuelig og foregår i modsætning til databasesoftware i følgende trin:



Leksikografen har som første redskab en XML-editor (eller en almindelig editor). XML-dokumentet fremstilles altså med et meget enkelt redskab. Grafisk orienterede XML-editorer som f.eks. XML SPY tilbyder her til hvert XML-dokument en trævisning, som er fuldt på højde med de grafiske databaseprogrammer (som f.eks. Micosoft Access). Editoren kan i sin basale form dog ikke undersøge, om XML-dokumentet svarer til en overordnet strukturdefinition (DTD). Dertil bruges parseren, en software, som meget vel kan være integreret i editoren. En kendt XML-parser følger med MS Internet Explorer 5.x og bliver brugt til en

trælignende visning af XML-filer i dette program. Parseren tilbyder alt efter tilgangen til XML-dokumentet forskellige definerede interfaces (API), som regel efter en af de to standarder fra W3C "SAX" og "DOM". Igen er parseren ikke et produkt, som viser XML-filen! Visningen af dataene i f.eks. en grafisk ramme (f.eks. browser) er ikke en opgave af XML men derimod af den software, som bygger i forskellige trin på en XML-parser. XML-processoren er en software, som benytter sig af parseren og som bearbejder dens output – f.eks. til en trælignende visning i browseren. Særlig interessant bliver denne opbygning i lag på det punkt, hvor XSLT-processoren muliggør udvælgelsen af bestemte informationsenhender fra en XML-fil baseret på en brugerdefinition. I den leksikografiske produktionsproces vil her være muligheden for at skabe leksikografiske opslagsværker med et datagrundlag og mulighed for mange forskellige "intendierte Benutzer" (Geeb 1997, Geeb 1998: 43ff.). Til den endelige grafiske gengivelse foreligger forskellige modeller. Kernen er her en webserver, som alt efter det valgte programmeringssprog benytter sig af en Java Servlet Engine eller andet software til afvikling og transformation af XML-filer til f.eks. HTML, som så sendes til brugerens webbrowser.

Leksikografen kan her i modsætning til de fleste databaseløsninger bearbejde de leksikografiske data (XML-filerne) forskellige steder i processen. XML-editoren vil på ethvert tidspunkt kunne bruges til bearbejdning af dataene. Parseren læser dataene ind og gemmer dem i hukommelsen alt efter den valgte metode i mange tilfælde som det XML-træ, som er kendt fra inputsiden i editoren. Allerede her er der ved hjælp af XML-processoren mulighed for at skrive information tilbage til datagrundlaget hhv. ændre det i ethvert henseende (alle knuder inkl. rodknuden). Bearbejdning af data kan altså foregå "on the fly" mens de udlæses og videregives til næste applikation. Yderligere kan de forandrede leksikografiske data skrives tilbage til det oprindelige datagrundlag. XML-processoren arbejder på dette punkt tæt sammen med parseren hvilket betyder, at denne bearbejdningsproces kan gentages cirkulært. Lignende forhold er der ved brugen af XSLT-processoren, som gennem XSLT tilbyder at modificere det oprindelige datagrundlag efter brugernes behov. Disse brugerbehov skal dog først være formuleret i et XSLT-Stylesheet. På grundlag af dette stylesheet kan XSLT-processoren danne et nyt datagrundlag (f.eks. en ny XML-fil) eller give disse bearbejdede og muligvis ny strukturerede data videre til en ny applikation. Som regel vil denne applikation være en Webserver, som så sender resultatet af XSLT-processorens arbejde til brugeren ved hjælp af f.eks. internettet. Dermed foregår i dette scenario transformationsprocessen på serversiden, dvs. arbejdet med at omdanne det oprindelige leksikografi-

ske dokument til det endelig viste dokument (på basis af brugerdefinitionen) foregår et centralt sted – hver gang dokumentet bliver sendt til en bruger. En anden mulighed ville være at sende brugeren såvel det leksikografiske datagrundlag (XML-filen) samt burgerdefinitionen (XSLT-Stylesheet), som så af software hos brugeren omdannes til det ønskede dokument. Browsere som f.eks. MS Internet Explorer > 5.5 samt Netscape Navigator > 6.1 har denne mulighed, dog kun med indskrænkninger og for Microsofts vedkommende med mange specielle tiltag, som ikke kan føres tilbage til de oprindelige og almengyldige standarder.

3. Muligheder for leksikografien

Leksikografens særlige muligheder ved brugen af XML til leksikografiske data (og informationer!) er mangfoldige. Teknologien er i dens basale form (XML, parsere, processorer, XSLT samt editorer) stabil og kan anvendes allerede i dag. Alene i den tid fra fremstillingen af denne artikel til trykken af publikationen vil der foreligge yderligere software og tiltag. I det følgende vil der blive nævnt de muligheder, som allerede i dag i det mindste lader sig beskrive ganske klart eller som endda kan bruges med det samme.

3.1 Projektsamarbejde i netværk

Netværkssamarbejde mellem leksikografer bliver meget nemmere ved brug af XML. Efter definitionen af en fælles struktur (DTD) kan de forskellige datafiler produceres samt offentliggøres decentralt. Hovedkravet er derefter kun, at alle projektmedarbejdere har adgang til samme netværk (i dag typisk internettet). Præsentationen af de leksikografiske data vil derimod på brugersiden være ensartet gennem centralt definerede XSLT-Stylesheets.

3.2 Individualisering og globalisering i netværkssamarbejde

XML har gennem såkaldte "namespaces" muligheden, at individuelle dokumenttypedefinitioner kan forsynes med et individuelt mærkat. Denne autorrelaterede information knytter DTD'en samtidig til en Uni-

form Ressource Identificator (indtil videre kun en webadresse). DTD'er fra leksikografiske projekter, enkeltpersoner eller store forlag vil dermed kunne bruges overalt i verdenen, forudsat at de er knyttet til internettet. Samtidig vil de ved rette brug af namespaces altid kunne føres tilbage til den rette ejer. Individualiseringen af leksikografiske modeller er her mulig samtidig med at de – forudsat at andre leksikografer vil bruge dem – kan have en global virkning.

Denne blanding af en individuel og samtidig almen synlige form for strukturdiskussion gælder også for XSLT-Stylesheets (som jo ligeledes er XML-dokumenter). Dermed kan XSLT-Stylesheets i deres anvendelse som leksikografiske brugerdefinitioner ligeledes forsynes med en globalt gældende autorinformation samtidig med at de er tilgængelige for alle interesserede leksikografer – og brugere af leksikografisk information!

3.3 Brugerdefinition gennem XSLT

Som antydet før er XSLT-Stylesheets det værktøj i XML-universet, som muliggør udvælgelsen af informationer på forskellig vis. Definitionen af de data, som skal udvælges fra det oprindelige datagrundlag, samt deres sortering og/eller bearbejdning svarer til en brugerdefinition i leksikografien. Hver bruger af et leksikografisk opslagsværk ("intendierter Benutzer") vil dermed få sig eget XSLT-Stylesheet, som sørger for den rette information i den ønskede rækkefølge og præsentation. Brugerdefinitionen på denne måde er variabel, den kan ændres fortløbende og er ikke knyttet til et program eller en software. XSLT-Stylesheets med den leksikografiske brugerdefinition ville kunne bruges overalt i verdenen med alle XSLT-processorer, som overholder XSL-standarden, lige meget om det er i Japan med en computer af mærke Macintosh eller i Italien med styresystemet Linux. – Begge dele, altså muligheden for at udarbejde denne brugerdefnition i tæt relation til de leksikografiske data samt muligheden for at genbruge den uafhængig af specielle softwareversioner, leverandører eller styresystemer må betegnes som et stort fremskridt.

3.4 Præsentation af leksikografiske informationer

Leksikografiske informationer kan ved brug af XML som datagrundlag nemt videregives til brugeren på mange forskellige måder. – XML-kilden kan udleveres til burgeren til videre brug f.eks. for at tilføje yderli-

gere oplysninger. XML filen kan omdannes og viderebearbejdes til alment visbar HTML ved hjælp af XSLT. Gennem brug af XSL-FO (Formatting Objects) kan XML-kilden omdannes på grundlag af et XLS-Stylsheet til bl.a. til det meget brugte, grafiske filformat Adobe PDF. På samme vis ville der kunne produceres andre printbare postscriptformater. Også brugen af disse data til WML/WAP og dermed præsentation af leksikografiske data på håndholdte enheder som mobiltelefoner er nemt muligt uden at der skal foretages ændringer i datagrundlaget eller allerede definerede XSL-Stylesheets og brugerdefinitionen. Hvis de leksikografiske XML-filerne desuden kobles med XML-applikationen SMIL (Synchronized Multimedia Intetration Language), kan de leksikografiske data indeholde alle kendte multimedia-elementer så som film, lyd, billeder etc. – og alt dette i et format, som kan transporteres over de kendte TCP/IP-netværk (f.eks. internettet) uden større downloadtider.

3.5 Brugerens indflydelse

I de kendte leksikografiske opslagsværk – elektronisk eller på papir – er der ikke de store muligheder for at strukturere informationerne på ny gennem en af brugeren selv oprettet brugerdefinition. Med XML i leksikografiske sammenhænge ville det være muligt, at give brugeren alle de redskaber (XSLT-stylesheet), som åbner for en fuld brugerdefinieret informationspræsentation. For brugerens vedkommende forudsætter det enten et kendskab til XML og XSLT, eller at der stilles en software til rådighed, som f.eks. gennem besvarelse af relevante brugerrelaterede spørgsmål (situation, intention, forudsætninger; Geeb 1998:39 ff) at danne denne XSLT-stylesheet. Brugeren får dermed en helt ny og meget udvidet rolle i den leksikografiske proces. Kvalitetsbedømmelsen af leksikografiske opslagsværker bliver som følge heraf mere individualisieret end hidtil kendt.

3.6 De klassiske strukturbegreber i leksikografien

De klassiske strukturbegreber som mikrostruktur og makrostruktur vil få en helt ny betydning ved brugen af XML som leksikografisk redskab. På den ene side vil brugerens store indflydelse på præsentationen og sammensætningen af informationerne muliggøre vedkommende at skabe en absolut brugerrelateret makro- og mikrostruktur. På den anden

side vil produktionen af leksikografiske data være helt adskilt fra den klassiske tanke om makro- og mikrostrukturen. Informationerne kan gennem XML-processoren og XSLT-processoren blandes og bearbejdes vilkårligt af alle brugere og leksikografer med adgang til de leksikografiske data. Mange af de traditionelle undersøgelser til makro- og mikrostruktur – både i metaleksikografien samt også i ordbogskritikken – skal tænkes igennem på ny, hvis de skal kunne bruges i denne sammenhæng.

3.7 Information i et netværk

Informationer i netværk kan i dag linkes sammen gennem HTML-links. Dette linkkoncept er kun en del af den omfattende URI-idé, som i sin gennemførelse muliggør at identificere ressourcer entydig på nettet uafhængig af deres placering, tid eller sted. Dette mål ligger udenfor XML, men med linksproget XLink og dokumentlinksproget XPointer står der et mangfoldigt linkredskab til rådighed. Leksikografiske data kan – afhængig af den software, som omsætter det leksikografiske XML-dokument – ved hjælp af XLink og XPointer linkes sammen på elementniveau uden at link-målet skal være forsynet med en tilsvarende markering. Desuden vil der ud fra en henvisning i leksikografiske sammenhæng være mulighed for at tilbyde brugeren flere linkmål – enten til frit valg eller også relateret til brugerens profil (gennem en XSLT-Stylesheet).

3.8 Universalisering af tegnsystemet

Del af XML-verdenen er det på to byte baserede tegnsystem UNICODE ("UTF-8" samt "UTF-16"). Begrænsningen på 127 eller i bedste fald 256 forskellige tegn (ASCII/ANSI) i computeren er dermed ophævet, da Unicode kan rumme op til 65.535 forskellige tegn. Det fremtidige tegnsæt kan om end ikke alle så dog vise de fleste af verdens (!) tegn. Med tegnsættet UCS-4, som er XML's fremtid, vil der være mere end 2 billioner forskellige tegn til rådighed. For leksikografien betyder det at konverteringsproblemer af tegn mellem sprog vil høre fortiden til.

3.9 Versionskontrol

Med XML er det nemmere og mere overskueligt end i en ren databaseløsning at forsyne leksikografiske data med oplysninger om ændringer i

indholdet. Denne del af leksikografens opgave kan nemt tildeles XML-processoren, hvis det ikke allerede er implementeret i en tilsvarende editor. Versioneringsdata følger dermed den enkelte leksikografiske oplysning, også ved brug i distribuerede systemer. Netop ved større projekter eller udveksling af data i netværk er det meget vigtigt, at de ved versionering opståede data gemmes sammen med det resterende datagrundlag, altså som regel sammen med lemmaet.

3.10 Datamanipulation i XML

Gennem den meget åbne struktur i den lagdelte XML-produktionsmodel er det for leksikografen muligt, at påvirke dataene flere steder i systemet. Det kan ved siden af almindelig produktion og bearbejdning også være under brugen af dataene i en parser. Ved denne lejlighed kunne man f.eks. forestille sig en automatiseret oversættelse af betydningsoplysninger eller en automatisk generering af en til lemmaet tilhørende fleksionsmodel. – Desuden er der jo omfattende muligheder for bearbejdning af dataene gennem XSLT og hermed på et tidspunkt, hvor de sendes til brugeren eller til en anden applikation.

4. Konklusion

XML i leksikografien byder på mange forbedringer og forenklinger, som desuden ligger helt på linie med tidens udvikling af informationssystemer (internettet). Mange af de nye muligheder for leksikografien ved brug af XML-familien er endnu ikke implementeret, men de ligger lige forude og befinder sig i et defineret område. Dog er der ligeledes stadigvæk områder, som er mere åbent. Søgningen i store dokumentmængder kan f.eks. være et performanceproblem, men det er et område, som vil blive løst udenfor leksikografiens rammer.

Skal leksikografen fremover være udvikler, programmør eller IT-ekspert for at kunne benytte sig af denne teknologi eller endda udøve leksikografien? Svaret må samtidig være ja og nej. En leksikograf skulle på baggrund af sin viden om informationsstrukturering nemt kunne klare at omforme en ordbogsartikel og dens mikrostruktur i en XML-fil. Også produktionen af en tilhørende datastrukturdefinition (her altså en DTD) skulle være muligt om end let ved brug af den fornødne software. Brugen eller programmeringen af en XML-parser, en XML-processor, en XSLT-processor eller en webserverinterface vil nok blive forbeholdt leksikografer med særlig interesse for og viden indenfor informationsteknologi. Det må være disse leksikografers opgave at skabe de nødvendige rammer (software), som integrerer de nævnte lag i

brugen af XML som datagrundlag i en brugervenlig enhed. Dog vil der altid være behov for, at leksikografen forstår den grundlæggende idé med XML indenfor leksikografien og med XML alment. Her er der ingen forskel fra datalagring og informationsstrukturering i XML og i databaser. Også ved brug af relationale (eller andre) databaser er der i det mindste en grundlæggende viden omkring datamodellering samt normalisering uundgåeligt. Mangler denne grundlæggende forståelse og viden – enten ved brug af XML eller ved brug af databaser – opstår alt for tit rene ordlister skrevet i et tekstbehandlingsprogram hvis værdi og nytte ville være langt større, hvis de var tilgængelige i en velovervejet struktur.

5. Litteratur

5.1 Links

Extensible Markup Language (XML) 1.0 (Second Edition). W3C Recommendation 6 October 2000: http://www.w3.org/TR/2000/REC-xml-20001006

Namespaces in XML. World Wide Web Consortium 14-January-1999: http://www.w3.org/TR/1999/REC-xml-names-19990114/

LeXeML. Lexicographic Markup Language: http://www.lexeml.org

Synchronized Multimedia Integration Language (SMIL). 1.0 Specification. W3C Recommendation 15-June-1998: http://www.w3.org/TR/REC-smil/

Synchronized Multimedia Integration Language. (SMIL 2.0) Specification. W3C Proposed Recommendation 05 June 2001: http://www.w3.org/TR/smil20/

XML Linking Language (XLink) Version 1.0. W3C Proposed Recommendation 20 December 2000: http://www.w3.org/TR/2000/PR-xlink-20001220/

XML Path Language (XPath). Version 1.0. W3C Recommendation 16 November 1999: http://www.w3.org/TR/xpath
XML Pointer Language (XPointer) Version 1.0. W3C Last Call Working Draft 8 January 2001: http://www.w3.org/TR/2001/WD-xptr-20010108/
XSL Transformations (XSLT). Version 1.0. W3C Recommendation 16 November 1999. http://www.w3.org/TR/xslt

5.2 Anden litteratur

Chen, Cindy Xinmin & Ashok Malhotra 2000: *XML queries via SQL* . Yorktown Heights, NY: IBM Watson Research Center
DuCharme, Bob 1999: *XML: the annotated specification*. Upper Saddle River, NJ [u.a.]: Prentice Hall PTR.
Geeb, Franziskus 1997: Die Benutzertypologie als Grundstein terminologischer und lexikographischer Arbeit, in: *Proceedings from XXII International Association Language & Busniess Conference 'Language and Business Life' ed. by Annelise Grindsted*, Vol. 2. Duisburg, 215–235.
Geeb, Franziskus 1998: *Semantische und enzyklopädische Informationen in Fachwörterbüchern. Eine Untersuchung zu fachinformativen Informationstypen mit besonderer Berücksichtigung wortgebundener Darstellungsformen.* Århus: Wirtschaftsuniversität Århus.
Goetz, Frank 2000: *SMIL. Multimedia im Internet mit Realsystem G2*. München [u.a.]: Addison-Wesley.
Goldfarb, Charles F. 1990: *The SGML Handbook.* Oxford: Clarendon Press.
Goldfarb, Charles F. & Paul Prescod 1999: *XML-Handbuch. Muenchen [u.a.]* : Prentice Hall
Harold, Eliotte Rusty 1998: *XML. Extensible Markkup Language*. Foster City: IDG-Books Worldwide.
HSK 5 = *Wörterbücher. Dictionaries. Dictionnaires. Ein internationales Handbuch zur Lexikographie.* Hrsg. von Franz Josef Hausmann, Oskar Reichmann, Herbert Ernst Wiegand, Ladislav Zgusta. Berlin/New York: de Gruyter 1989–1991.
Kay, Micael (2000): XSLT. Programmer's Reference.Girmingham: Wrox Press.
Lobin, Henning 1998: *Informationsmodellierung in XML und SGML*. Berlin: Springer.
Michel, Thomas 1999: *XML kompakt. Eine praktische Einführung*. München, Wien: Hanser.

Möhr, Wiebke und Ingrid Schmidt (Hrsg.) 1990*: SGML und XML. Anwendungen und Perspektiven*. Berlin: Springer.

Pitts-Moultis, Natanya; Cheryl Kirk 1990: *XML Black Book*. Scotsdale (AZ): Coriolis.

Wiegand, Herbert Ernst 1989: Aspekte der Makrostruktur im allgemeinen einsprachigen Wörterbuch: alphabetische Anordnungsformen und ihre Probleme; in: *HSK 5.1*, 719–.

Wiegand, Herbert Ernst 1989a: Der Begriff der Mikrostruktur: Geschichte, Probleme, Perspektiven. in: *HSK 5.1*, 409–462.

Wiegand, Herbert Ernst, Henning Bergenholtz & Sven Tarp 1999: Datendistributionsstrukturen, Makro- und Mikrostrukturen in neueren Fachwörterbüchern. In: *Fachsprachen. Languages for Special Purposes. Ein internationales Handbuch zur Fachsprachenforschung und Terminologiewissenschaft* […]. Hrsg. v. Lothar Hoffmann, Hartwig Kalverkämper, Herbert Ernst Wiegand. 2. Halbbd. Berlin. New York 1762–1832.