Semantiske netteknologier for formidling av humanistisk forskningsmateriale

XML Topic Maps som publiseringsverktøy for humaniora

Trond Karl Pettersen

Mastergradsoppgave

Seksjon for humanistisk informatikk
Universitetet i Bergen

Februar 2006

Abstract

Semantic Web technologies for publication of humanistic research data. XML Topic Maps as a publishing tool for the humanities.

Part 1 (chapters 1 through 3) of this master's thesis consists of a critical discussion of why and how to represent knowledge on the World Wide Web. It starts by giving an overview of the historical development of networks, network technologies and the World Wide Web. Thereafter, insight from the field of Artificial Intelligence is discussed, focusing on the limitations of micro worlds and the separation of algorithm and problem-domain. Knowledge drawn from AI may be seen as converging with that of network technologies and distributed processing, giving us a vision of a Semantic Web where sets of data are self-describing - beyond the context of layout. We define the technologies of such a Semantic Web to be based on four main components: unambiguous identity-criteria, classes, objects, and relations. Hence, Semantic Web technologies and standards must be able to convey information about ontologies, and will typically take the form of associative 'information overlay'-technologies such as W3C's Web Ontology Language (OWL) and ISO's Topic Maps.

Part 2 discusses an implementation of a Semantic Web technology. The Web application[1] developed as part of this project is used for publication of research data from the project "Det vestnorske hellerprosjektet" at the Department of Archeology, University of Bergen. The project's findings will be published using XML Topic Maps, which allows us to navigate the information structure based on the semantics of the information model itself.

This report is written in Norwegian.

Keywords: Semantic Web, Topic Maps, XTM, XML, RDF, OWL.

[1] URL: `http://huin.uib.no/hellerprosjektet/`.

Forord

Denne oppgaven er utarbeidet i forbindelse med Seksjon for humanistisk informatikks deltakelse i Det vestnorske hellerprosjektet ved Arkeologisk institutt, UiB. Takk til Hellerprosjektet for interessante arbeidsoppgaver, og lykke til videre.

Jeg vil også uttrykke takknemlighet overfor Guro for kommentarer og tålmodighet - spesielt i innspurten. Nå kan endelig jeg lese korrektur for deg.
Mange takk til lektor Knut Pettersen for korrekturlesing og kommentarer.

Til slutt vil jeg takke min veileder, Audun Stolpe, for konstruktiv kritikk og meget fruktbar veiledning.

Bergen, 1. februar 2006
-Trond K. Pettersen

Made with LaTeX 2$_\varepsilon$

Innhold

Figurer

Kapittel 1

Innledning

Det vestnorske hellerprosjektet (HP) er et tverrfaglig forskningsprosjekt ved Universitetet i Bergen (UiB), hvis hovedmålsetning er å "belyse endringer i erverv, sosial identitet og rituell aktivitet i hellerne på kysten av Norge i steinalder, bronsealder og eldre jernalder" (Bergsvik et al. 2004, s.1). Prosjektet ble påbegynt høsten 2004, med planlagt avslutning høsten 2007. Målsetninger og problemstillinger gitt i Bergsvik et al. (2004) skal belyses gjennom nye arkeologiske undersøkelser i hellere på Vestlandet, samt analyser og re-publisering av resultater fra tidligere utgravinger. Deltakere fra flere ulike fagretninger bidrar til prosjektet, og inkluderer blant annet Bergen Museum, Nordisk institutt og Botanisk institutt ved UiB, Radiologisk laboratorium ved NTNU, Trondheim, samt Seksjon for humanistisk informatikk, UiB.

1.1 Problemområde

Humanistisk informatikk sin rolle i Hellerprosjektet er å "forsøke å utvikle et rammeverk for web-formidling som en integrert del av forskningsprosessen" (Bergsvik et al. 2004, s.1), noe som er ment å komplementere (populærvitenskapelig) bokformidling av prosjektet og en utstilling ved Bergen Museum. Et slikt rammeverk for web-formidling skal ta form av et nettsted basert på den internasjonale emnekartstandardens XML-syntaks; XML Topic Maps (XTM). Bruk av emnekart begrunnes av Bergsvik et al. (2004) gjennom at emnekart er en standard for "klas-

sifisering, gjenfinning, utveksling og presentasjon av informasjon og kunnskap". I prosjektbeskrivelsen til Hellerprosjektet går det videre frem at nettstedet skal baseres på XTM, fordi en antar at XTM gir oss nye muligheter "for navigering og søking i informasjonsmengden" (Bergsvik et al. 2004, s.7).

1.1.1 Problemstilling

Denne mastergradsavhandlingen har utgangspunkt i et behov for nettpublisering av informasjon fra Det vestnorske hellerprosjektet. I tråd med Hellerprosjektets prosjektbeskrivelse (Bergsvik et al. 2004) er det utviklet en vevapplikasjon basert på den internasjonale emnekartstandardens XML-syntaks; XML Topic Maps. Arbeidet er ikke ment å være en endelig løsning for Hellerprosjektet, men det skal være mulig å benytte applikasjonen for publisering av forskningsdata ved hjelp av XTM.

Den overordnede problemstillingen for arbeidet presentert i denne rapporten er:

> Undersøke i hvilken grad en betydningsorientert og assosiativ informasjonsturktur som emnekart egner seg for formidling av forskningsmateriale innenfor humaniora, i særdeleshet arkeologi.

Problemet slik det er presisert ovenfor skal utforskes gjennom utvikling av et XML Topic Maps-basert nettsted, samt en kritisk diskusjon av bakenforliggende teori. Oppgaven er med andre ord til en viss grad todelt; i den ene delen har det vært utført et praktisk arbeid med utvikling av et XTM-basert nettsted, mens den andre delen (denne rapporten) diskuterer bakenforliggende teori og relatert teknologi, samt gir en evaluering av XTM og resulterende nettsted som et rammeverk for formidling av forskningsmateriale innen humaniora.

For nettstedet utviklet i oppgavens praktiske del gjelder følgende krav:

- nettstedet skal konstrueres for formidling av forskningsmateriale ved hjelp av XTM.
- skal være dynamisk bygget opp gjennom at endringer i XTM gjenspeiles i informasjon som presenteres for brukere.

- bør enkelt kunne utvides og bli dynamisk med hensyn til flerspråklighet.

- bør enkelt kunne utvides med tanke på tilpasning av generert kode for ulike typer klienter[1] - eksempelvis mobiltelefoner, som krever mindre informasjon per side grunnet lite display, versus for eksempel PC, etc.

Den teoretiske delen av oppgaven vil fokusere på de teoretiske spørsmålene, heller enn implementasjonsmessige problemer (som bare kort vil nevnes). Dette gjelder blant annet metoder for å representere mening i informasjon merket for datamaskinell prosessering.

1.1.2 Avgrensing

Ingen prosjekter kan favne om alt, og ettersom utvikling av en vevapplikasjon i seg selv er en tidkrevende prosess, som på dette nivået ikke nødvendigvis gir mye ny innsikt, har det også for denne oppgaven vært viktig å sette begrensninger for hva som skal gjøres. Det har blant annet ikke vært mulig å fullt ut implementere alle aspekter ved vevapplikasjonen som i utgangspunktet ville vært 'kjekke å ha'. Dette gjelder blant annet flerspråklighet, samt fastslåing av type klient og klient-tilpasset presentering av informasjon. Andre dynamiske aspekter som ikke har vært prioritert gjelder filtrering av informasjon - for eksempel for ulike typer brukere - basert på de muligheter XTM gir oss. Som nevnt ovenfor skal disse funksjonene kunne legges til, da for eksempel filtrering for så vidt er en viktig del av XTM, men slike 'tekniske aspekt' ligger litt utenfor perspektivet for oppgaven. Det er heller ikke tatt høyde for at alt skal være 100% klart for bruk i et produksjonsmiljø, noe som blant annet gjenspeiles i design, brukervennlighet[2] med mer. For forslag til mulige forbedringer av den utviklede applikasjonen, se oppgavens avsnitt 4.2.4.

[1] En klient betyr her et dataprogram som sender en forespørsel til, eller benytter seg av, et annet dataprogram. En klient kan for eksempel være en nettleser som sender en forespørsel etter dokumentet `/hellerprosjektet/index.spy` til/på tjeneren (eng. "server") som befinner seg bak `huin.uib.no`.

[2] Her har det vært lagt størst vekt på å fremheve de ulike bestanddelene (mest av alt relasjoner) i et emnekart, ikke på hvordan elementene presenteres for brukeren. Dette gjør at nettstedets sider per dags dato ikke er optimalisert for alle tenkelige brukergrupper, men forhåpentligvis vil det fungere for hovedmålgruppen - arkeologer.

Det ville selvsagt vært interessant og undersøkt hvordan brukere oppfatter vevapplikasjoner som vektlegger assosiativiteten i informasjonsmengder. Man kunne her utført en brukerundersøkelse med hensyn til oppbygningen av et nettsted og de ulike navigasjonsmuligheter som XTM gir oss, slik at en bedre kunne fastslått hvilken nytteverdi teknologier som XTM har på dette punktet, rent teoretisk, eller også praktisk, og hvordan brukere tenker omkring dette. Da en kvantativ / kvalitativ brukerundersøkelse ville vært en tidkrevende prosess i seg selv, og utelukket en behandling av andre teoretiske aspekter som for meg syntes mer interessante, har jeg heller ikke kunnet inkludere en slik undersøkelse i prosjektet.

Når det gjelder de teoretiske delene av oppgaven søker ikke oppgaven å gi et endelig svar på eksempelvis det meget omfattende problemet med fletting av informasjon, men forsøker å trekke enkelte konklusjoner gjennom å sammenligne emnekart og forskning på områder som ontologier og kunnskapsrepresentasjon.

1.2 Definisjoner

Mange av de begreper som benyttes i denne oppgaven kan ha ulike betydninger. Dette avsnittet lister definisjoner av ulike sentrale begreper (slik de er brukt her) som gjerne ikke diskuteres videre, men som allikevel ligger til grunn for teksten.

Kunnskapsrepresentasjon (KR) defineres gjerne som "kunnskap organisert og representert i et formelt språk for bruk av datamaskiner" (Dictionary.com 2005).

En *mengde* defineres i mengdelære som en samling elementer. Et element kan her være hva som helst, også andre mengder (Haggarty 2002). Eksempler på mengder er {3,2,8} (elementene i mengden er her '3', '2' og '8'), {arkeologi, Per, 54}, {kniv, øks, pil, pren}. Den innbyrdes rekkefølgen til elementene i en mengde er uvesentlig; {1,2,3} betegner den samme mengden som {3,1,2}. At et element a er en del av en mengde A angis med tegnet $\in$ ("i") - for eksempel Beinpiler $\in$ AllePiler. For store mengder er det ugunstig, eller umulig å angi hvert enkelt element i mengden, og en mengde S hvor predikatet P gjelder for hvert element x i S angis med mengdenotasjon; $S = \{x\colon P(x)\}$ som leses "x, slik at P gjelder for alle x" (Haggarty 2002). For eksempel angir $S = \{x\colon x$ er et steinredskap$\}$ mengden av steinredskap

- x er et element i S hvis og bare hvis x er et steinredskap.

Merkespråk (eng. 'Markup Language') er sett av symboler og regler for å merke informasjon (Dictionary.com 2005). Ved hjelp av 'merkene' i merkespråk kan en angi hvilken type informasjon en gitt informasjonsbit er av. HyperText Markup Language (HTML) er et kjent eksempel på et merkespråk. HTML er et såkalt *prosedyre-orientert merkespråk* (Pitti 2004), da det (i dagens form) angir hvordan programmer (for eksempel nettlesere) skal presentere innholdet i et dokument for en menneskelig leser. XML Topic Maps er et eksempel på et *deskriptivt merkespråk* som sier noe om strukturen til, eller logikken i, informasjonen i et dokument.

Metadata er data om data (Dictionary.com 2005). "Metadata" eller "metainformasjon" brukes gjerne som en betegnelse for informasjon om objekter (Garshol 2004). Informasjon om forfatteren av denne oppgaven, for eksempel forfatterens navn, er et eksempel på metainformasjon om oppgaven. Metadata er data i seg selv, og skillet mellom når data er metadata eller ikke ligger i hvordan dataene brukes.

Semantikk er meningsteori; teori om meningsbevaring. For datamaskinell bruk kan det være hensiktsmessig å definere ulike grader av "maskin-tilgjengelig" semantikk, fra implisitt semantikk - for eksempel forfatterens tanker om innholdet i en roman - til eksplisitt formell semantikk (for bruk av datamaskiner) (Uschold 2001). Se også avsnitt 3.1 på side 24.

Tekstmerking (eng. 'Markup') er tekstbiter lagt til data/informasjon i et dokument for å kunne si noe om merket informasjon (Dictionary.com 2005). For eksempel kan en merke tekstbiten 'En adresse' for bruk av et program ved å legge til merking á la <address>En-adresse</address>, adresse: En-adresse, eller lignende.

Vevapplikasjon er for denne oppgaven definert som et program, eller en samling skript som benyttes for å løse et problem på WWW. Eksempler: system for publisering av informasjon, on-line diskusjonsforum, etc.

World Wide Web (WWW) består av mengden ressurser på Internett som er tilgjen-

gelig via protokollen HyperText Transfer Protocol (HTTP), mens *Semantic Web* (SW) er en visjon om en web hvor ressurser er deskriptivt og eksplisitt, formelt semantisk merket etter klasser og identitet. SW er ikke ment å erstatte WWW, men å bygge på WWW ved å blant annet tilføye metadata. Se også avsnitt 3.1 på side 24.

1.3 Oppgavens oppbygning

Tekstens del 1, kapittel 1-3, diskuterer teori bak en semantisk verdensvev hvor informasjon er lagret i selvbeskrivende datasett for datamaskinell prosessering utover layoutkontekst. Del 2, kapittel 4, diskuterer en praktisk implementasjon av en semantisk netteknologi.

Mer spesifikt gjør kapittel 2 rede for utviklingen av nettverk og nettverksteknologier. Herunder faller Internett, Vannevar Bushs tanker om assosiativ indeksering, Tim Berners-Lees World Wide Web, distribuert prosessering, samt svakheter ved HTML. Deretter diskuteres relevant innsikt fra kunstig intelligens (KI), med vekt på begrensninger ved mikroverdener og separering av algoritme og problem for representering av kunnskap. I avslutningen av kapittel 2 ser vi hvordan det i de senere år har oppstått et liknende behov for kunnskaprepresentasjon på WWW.

Kapittel 3 diskuterer kriterier for en semantisk verdensvev hvor datasett i større grad enn på dagens WWW er selvbeskrivende. Det gis her en formålstjenelig definisjon av semantiske netteknologier, hvoretter det gjøres rede for ontologier, ontologiutvikling og organisering av ontologisk kunnskap. Før oppgavens kapittel 4 gis det en kortfattet presentasjon av noen betydningsbevarende merkespråk.

Kapittel 4 diskuterer implementasjon av en semantisk netteknologi; XML Topic Maps for Det vestnorske Hellerprosjektet. Kapittelet gir først en oversikt over emnekartstandarden, før det gjøres rede for valg tatt under utvikling av vevapplikasjonen. Deretter vises det gjennom eksempler hvordan en assosiativ informasjonsturktur kan gi opphav til navigering basert på informasjonsmodellens iboende assosiativitet. Avslutningsvis gjøres det opp noen tanker omkring forskningsformidling på Web.

Kapittel 2

Historisk bakgrunn

Ettersom nettverk og nettverksteknologier er en forutsetning for at datamaskiner skal kunne utveksle informasjon på Internett, gis det i dette kapittelet en komprimert oversikt over deler av den historiske utviklingen av nettverksteknologier. Deretter diskuteres World Wide Web og HTML, med sine styrker og svakheter, før det gjøres rede for relevante problemer innenfor området kunstig intelligens (KI). Kapittelet viser til slutt hvordan disse to teknologiene kan sies å konvergere og dermed danne grunnlag for en verdensvev hvor større deler av informasjonsinnholdet er tilgjengelig for datamaskinell prosessering.

Ideelt sett hadde den historiske oversikten bygget på flere kilder, men da det ikke lyktes forfatteren å finne ytterligere (god) litteratur knyttet til overgangen fra spesialiserte til distribuerte systemer, er kapittelets første del i hovedsak basert på Naughton (1999).

2.1 Nettverk

De første datamaskinene var svært forskjellige fra dagens digitale datamaskiner. Ikke bare var maskinene svært kostbare (fra $500.000 USD) og plasskrevende (eksempelvis ca. 2500 kvadratfot og 250 tonn), de var også analoge og måtte omprogrammeres for hvert nye problem som skulle løses (Naughton 1999). Maskinene ble ikke produsert i tusentall på løpebånd, hver datamaskin var unik og programvare utviklet for en datamaskin kunne ikke benyttes av andre datamaskiner. Mange av maskinene var dessuten konstruert på en måte som gjorde at det til enhver tid bare kunne være en bruker tilknyttet hver maskin, og kunne ikke dele prosesser mellom brukere, eller program. Selv om maskiner

allerede fra relativt tidlig av kunne motta data fra eksterne terminaler var det heller ikke mulig å koble dem sammen i nettverk, da analoge signaler lett korrumperes over avstand. Samtidig gjaldt det at maskinene ikke kunne kommunisere seg imellom, grunnet ulikheter i både arkitektur og programvare (Naughton 1999).

På begynnelsen av 1960-tallet anslo så Paul Baran ved RAND (USAs 'think-tank' under den kalde krigen) at det måtte være mulig å flytte meldinger mellom datamaskiner på hver side av USA ved å sende dem i form av digitaliserte beskjeder over et distribuert nettverk av noder (datamaskiner). Analoge signaler kunne ikke benyttes ettersom kvaliteten på analoge signaler synker etter hvert som avlagt distanse øker, og nettverket måtte være distribuert fordi et sentralisert nettverk (bygget opp rundt en, eller noen få sentrale noder) ville ha vært svært sårbart i en krisesituasjon - dersom de sentrale nodene feilet, ville hele nettverket feile (Naughton 1999). Ettersom de mellomliggende nodene også måtte kunne utføre feilsjekking og sende en digital melding videre i nettverket, måtte hver node være en digital datamaskin. Baran viste også at det måtte være en viss overflod av data sendt over et slikt nettverk - for å sikre at informasjon ikke gikk tapt. Han antok videre at hver melding kunne deles opp i biter og sendes "stykkevis" over nettet for til slutt å bli satt sammen i rett rekkefølge på målmaskinen (Naughton 1999).

Briten Donald Davies, som ikke kjente til Barans nettverk-teori, søkte i likhet med Baran å konstruere et nettverk som tillot kommunisering datamaskiner imellom - blant annet for å kunne utnytte de kostbare maskinenes prosesseringskraft bedre. Davies mente også at meldinger mellom datamaskiner optimalt burde sendes oppdelt i stykker (Naughton 1999). Disse stykkene kalte han pakker (eng. "packets"). Davies foreslo at hver pakke skulle være av en bestemt størrelse (1024 biter / 128 tegn) og ikke bare inneholde data, men også en "header" inneholdende informasjon om avsender, mottaker, en kontrollkode for å kunne sjekke hvorvidt pakkens data var blitt korrupt ('ødelagt', tapt informasjon) under overføring, og et sekvensnummer som anga pakkens nummer i sekvensen av pakker fra den fragmenterte originalmeldingen (Naughton 1999). Innføringen av pakker med headere tilsa at enhver node bare måtte kunne lese headeren og destinasjonsadressen, samt sende pakken videre til neste node. Informasjonen lagret i headerne gjorde det dessuten mulig for målmaskinen å sette sammen pakker mottatt i vilkårlig rekkefølge, samt sende bekreftende responser til avsendere. Davies foreslo dessuten et system for å koble flere brukere til et nettverk via mellomliggende datamaskiner (Naughton 1999). I 1966 presenterte så Davies sine tanker om et "packet-

switched"-nettverk, og i 1967 - etter at han var utnevnt som sjef for National Physical Laboratory (NPL) - satte han opp et team som planla å konstruere et Local Area Network (LAN) mellom ti datamaskiner og diverse annet datautstyr (Naughton 1999).

Samme år ble planene for USAs Advanced Research Projects Agencys (ARPA) nettverk, ARPANET, for første gang offentlig presentert. Dette skjedde på samme konferanse som Roger Scantlebury fra Davies' team introduserte en detaljert beskrivelse av NPLs nettverk, og Lawrence G. Robert fra ARPA returnerte fra konferansen overbevist om at Davies' 'packet-switching' var løsningen som ARPA trengte for å fullføre sine egne planer om et nettverk, som allerede var under utvikling (Naughton 1999). I Januar 1969 tegnet ARPA så kontrakt med firmaet BB&N for produksjon av "interface message processors"-er (IMP), en slags 'mellomstasjoner' (små datamaskiner) som skulle ta seg av kommunikasjon mellom ulike noder i et nettverk (Naughton 1999). Ettersom BB&N hovedsaklig konsentrerte seg om å få data transportert mellom IMPer, og ikke om løsninger utover dette kravet, nedsatte ARPA gruppen Network Working Group (NWG) som bestod av representanter - hovedsaklig mastergradsstudenter - fra ulike bidragsytere (Naughton 1999). NWG utformet Request For Comments (RFC)-systemet[1], et "publiseringssystem" som la vekt på samarbeid og utveksling av tanker og ideer knyttet til nettverk, og bidrog i sin tur med utvikling av protokoller[2], blant annet Network Control Program (NCP) som ble en viktig ingrediens i ARPANET (Naughton 1999).

2.1.1 Nettverk av nettverk - Internett

ARPANET ble i og for seg en suksess, men hadde flere svakheter. For det første krevde nettverket at alle datamaskiner benyttet samme programvare, og baserte seg på en sentral kontroll med IMPene (programoppdateringer, feilsøking, etc.). For det andre fantes det alternative systemer som baserte seg på totalt forskjellige virkemåter og protokoller (Naughton 1999). Både Frankrike og England hadde utviklet egne nettverk, og på Hawaii hadde University of Hawaii utviklet et nettverk - ALOHA - som benyttet radiobølger for å sende data mellom campuser spredt utover ulike øyer. Mens ARPAs NCP var bygget med en antagelse av at nettet var pålitelig, var for eksempel ALO-

[1] Samtlige RFC-notater / -rapporter fra 1969 og frem til i dag ligger offentlig tilgjengelig på Internett. Se for eksempel `http://www.ietf.org/rfc/rfc.html`.

[2] En protokoll er i denne sammenheng "et sett med standardiserte regler for utveksling av data mellom samvirkende, uavhengige systemer" (Hannemyr 1999, s. 11).

HA bygget på totalt motsatte antagelse - man måtte her anta at pakker ville kollidere under overføring, og at nettet av natur derfor var upålitelig (Naughton 1999). Med utgangspunkt i disse problemene begynte Bob Kahn ved ARPA å interessere seg for hvordan en kunne koble alle de ulike nettverkene sammen i ett stort Internett. Sammen med Vinton Cerf utviklet Kahn en ny ide og protokoll for utveksling av meldinger mellom datamaskiner. Protokollen ble hetende Transmission Control Protocol (TCP), og hovedtanken bak TCP var at den eneste jobben til 'mellomstasjonene' i et nettverk var å forstå protokollene benyttet av avsender og mottaker, samt å videresende pakker som var pakket inn i virtuelle konvolutter (Naughton 1999). Innholdet i pakker er for TCP likegyldig.

Etter ytterligere arbeid, som blant annet viste at TCP alene ikke var like godt egnet for alle typer nettverksbaserte tjenester og at det vanskelig ville la seg gjøre å bygge slik støtte inn i én protokoll, ble TCP senere delt inn i to ulike protokoller; en ny Transmission Control Protocol (TCP) og en Internet Protocol (IP) (Clark 1995). TCP håndterer pakker, mens IP lar en identifisere en maskin blant millioner av maskiner, og angir en standard metode for sending av data mellom to maskiner. I 1982 skiftet alle noder i ARPANET ut NCP med TCP/IP, og i 1985 ble TCP/IP bygget inn i operativsystemet UNIX (Naughton 1999). Ettersom TCP/IP støtter et vidt spekter av nettverksteknologier og stiller seg likegyldig med hensyn til type informasjon sendt over nettverkene (Clark 1995), var Internett med TCP/IP en realitet.

Selv om ARPANET linket sammen datamaskiner over hele USA (og deler av Europa), var det bare en akademisk og militær elite som var priviligert med tilgang til nettverket. Samtidig fantes det i andre grupper av samfunnet - for eksempel blant lærere, tidligere ARPA ansatte, datainteresserte, etc. - et stort ønske om å være tilknyttet et større nettverk (Naughton 1999). Dette ønsket ble til en viss grad oppfylt da Mike Lesk hos Bell Labs utviklet et UNIX program kalt 'UNIX-to-UNIX-Copy' (UUCP) for å automatisere innhenting av oppdaterte filer fra eksterne datamaskiner. UUCP gjorde det mulig for en UNIX-basert datamaskin å koble seg på en annen datamaskin, søke etter endringer i bestemte filer, og kopiere endringene over på eget lagringsmedium. Prosessen ble automatisert av et program kalt NetNews, som i 1980 ble skrevet om av to studenter og døpt Usenet News. Usenet News ble senere kjent som nyhetsgrupper; "news groups" (Naughton 1999).

Mens ARPANET var preget av eksklusivt medlemskap og Usenet var et program skrevet for UNIX-maskiner, begynte Ward Christensen og Randy Suess på slutten av

1970-tallet utviklingen av et "homebrewed Net" for overføring av filer mellom hjemmedatamaskiner (via telenettet) (Naughton 1999). Programmet som Christensen skrev ble gjort offentlig tilgjengelig for alle som ønsket å kommunisere via sine private datamaskiner (PC). Christensen skrev dessuten et program som gjorde det mulig å gjøre datamaskiner om til egne "store-and-forward"-systemer (Computer Bulletin Board System (CBBS)), mens Suess designet et enkelt "microcomputer-to-microcomputer"-system. Disse to elementene gjorde det (i teorien) mulig for alle med en PC å ringe opp og kommunisere med andre PCer og PC-brukere. Også resultatet av dette arbeidet ble gjort offentlig tilgjengelig. I 1980 utviklet så Tom Jennings et alternativt nettverk basert på arbeidet til Christensen og Suess; Fidonet[3] (Naughton 1999).

2.2 World Wide Web (WWW)

Til tross for systemer som ARPANET, Fidonet, Telnet, FTP, E-mail og tidlige hypertekstsystemer, manglet det på slutten av 1980-tallet / begynnelsen av 1990-tallet fremdeles en "samlende" applikasjon som muliggjorde gjenfinning av ulike ressurser fra ulike nettverk over en og samme protokoll.

Så tidlig som i 1945 hadde Vannevar Bush publisert en artikkel, *As We May Think*, hvor han beskrev et tenkt "minne-supplement"-system kalt Memex. Bush var bekymret over at mens de senere års vitenskap hadde gitt menneskeheten økt kunnskap, så var mengden av informasjon etter hvert blitt så overveldende stor at forskere i økende grad ikke hadde hverken tid eller hukommelse til å utnytte all eksisterende viten (Bush 1945). Svært mye verdifull forskingstid gikk tapt i manuell søken etter informasjon, og menneskeheten ville derfor nyte godt av et mekanisert arkiv hvor en lett kunne gjenfinne den typen informasjon som en var på jakt etter. Bush så for seg en maskin som tillot lagring av enorme mengder informasjon (på mikrofilm) - og ikke minst søking i lagret informasjon; "A library of a million volumes could be compressed into the end of a desk" (Bush 1945). Memex skulle dessuten lagre informasjon assosiativt, fordi menneksets tankeprosesser opptrer assosiativt: "The human mind [...] operates by association. With one item in its grasp, it snaps instantly to the next that is suggested by the association of thoughts" (Bush 1945) (å høre Beatles' sang Strawberry Fields Forever kan eksempelvis få en til å tenke på smaken av jordbær). Assosiativ indeksering var ifølge Bush en essensiell egenskap ved Memex (Bush 1945), som altså skulle tillate en å bevege

[3] Se `http://www.fidonet.org` for oppdatert informasjon om stadig aktive BBSer.

seg mellom fragmenter av informasjon som assosiativt hørte sammen. Memex ble aldri konstruert, men enkelte av dens prinsipper - for eksempel muligheter for assosiativ linking - ligger til grunn for blant annet hypertekster og WWW (Naughton 1999).

På slutten av 1960-tallet og begynnelsen av 1970-tallet ble de aller første hypertekst-systemene utviklet, og i 1980 utviklet Tim Berners-Lee programmet ENQUIRE (for "enquire within about everything"). Bakgrunnen var et behov blant forskerne på CERN-laboratoriet i Sveits, hvor Berners-Lee jobbet, for å lagre og overføre dokumenter på en måte som tillot enkel referering (til andre dokumenter), og ENQUIRE tillot oppretting av assosiasjoner mellom hva som helst (Naughton 1999).

Fra 1989 til 1991 utviklet så Berners-Lee merkespråket HyperText Markup Language (HTML) som en applikasjon av den internasjonale informasjonsorganiseringsstandarden Standard Generalized Markup Language (SGML). I tillegg til HTML utviklet han en protokoll - HyperText Transfer Protocol (HTTP) - som beskrev regler for overføring av HTML-dokumenter mellom ulike datamaskiner, en adresseringsprotokoll - Uniform Resource Locator (URL), samt demoversjoner av henholdsvis den førtste nettleseren og den første WWW-tjeneren (eng. "server") (Naughton 1999). Målet med HTTP og HTML var en enkel måte for overføring av ulike typer informasjonsressurser mellom forskningsinstitusjoner, samt utvikling av en enkel metode for linking mellom ulike ressurser i et nettverk. Dette for å gjøre informasjon lett tilgjengelig for akademikere spredt over et større geografisk område (Berners-Lee 1991). Prosjektet kalte han "World Wide Web" (WWW).

HTTP ble utviklet som en enkel protokoll som i utgangspunktet krever få prosesseringsressurser på tjener-siden, og ved hjelp av relativt enkelt utformete forespørsler og responser sendes data mellom klient (for eksempel en nettleser) og tjener. Første HTTP-versjon (HTTP 0.9) var særlig enkel, og en forespørsel fra en klient til en tjener var bare 1 linje lang[4]. HTTP 0.9 ble senere utvidet, men grunnprinsippene består, og en transaksjon på WWW består fremdeles av en HTTP-forespørsel (fra klient til tjener) og en HTTP-respons (fra tjener til klient) (Shiflett 2003). HTTPs enkle design, kombinert med enkelheten til HTML (over for eksempel SGML) og det faktum at Tim Berners-Lee - lik de fleste som bidro til Internett før ham - ikke patenterte sin oppfinnelse, har utvilsomt hatt mye å si for utvidelsen av verdensveven. Da Marc Andreesen i 1993 utviklet Mosaic, den første nettleseren med et grafisk brukergrensesnitt og mulighet for å se vise

[4]For eksempel `GET /index.html` som forespørsel (fra klient) for et dokument med navnet `index.html`.

bilder i HTML-tekster, eksploderte markedet (Hannemyr 1999). Mye takket være Mosaic og WWW gikk Internett i løpet av få år fra å være en samling mindre nettverk, for de få, til å bli et verdensomspennende gigantnettverk med millioner av brukere (Hannemyr 1999).

2.2.1 Nettorientert informasjonsbehandling

Distribuert prosessering

Tidlige datamaskinsystemer eksisterte altså som isolerte øyer hvor hver enkelt datamaskin tok seg av all prosessering av informasjon og kommuniserte med lokale brukere. Systemene var derfor svært spesialiserte, og informasjonen som ble prosessert var tilpasset spesiell programvare. Utviklingen av nettverksteknologier medførte dog, som vi har sett, at stadig flere datamaskiner gikk fra å være sentrum i egne univers til å bli en liten del av et, om enn ikke bokstavelig, uendelig nettverk av datamaskiner. Ved hjelp av de ulike nettverkene og protokollene (TCP/IP, HTTP) kunne digitale data nå flyttes mellom ulike enheter, og teknologier som Ethernet[5] gjorde at denne overføringen til dels kunne foregå i et imponerende raskt tempo.

Man fikk derfor i de nye teknologiene et system som tillater en å la en maskin A løse et problem og sende resultatene til en annen maskin B. B kan deretter inkludere resultatet fra A i sine videre beregninger. Nettverksteknologiene åpnet med andre ord for distribuert prosessering; ved å dele et problem inn i mindre enheter og utnytte prosesseringskraften til flere noder i et distribuert nettverk kan et problem løses på kortere tid enn hva som ville vært tilfellet dersom bare en enkelt datamaskin hadde måttet ta seg av all dataprosessering. Før utviklingen av digitale datamaskiner og pakkebaserte nettverk var distribuert prosessering i realiteten umulig, ettersom overføringstiden var høy (A ville måttet vente lenge på svar fra B) og datatapet stort (Naughton 1999).

Etter hvert som antallet nettverk tilknyttet Internett vokste og overføringshastigheter økte, ble det mulig å utnytte distribuert prosessering også på tvers av landegrenser og kontinenter. Et delproblem kan for eksempel sendes fra en datamaskin i USA til en datamaskin i Norge, som igjen kan returnere løsningen på problemet til opphavsmaskinen i USA. Dette utnyttes eksempelvis av SETI@home[6], hvor tusenvis av noder spredt

[5] Ethernet er en metode for hurtig overføring av store mengder data mellom ulike enheter i heterogene nettverk (datamaskiner, plottere, printere, etc.), utviklet ved Xerox Palo Alto Research Center (PARC) (Naughton 1999).

[6] `http://setiathome.berkeley.edu/`

over hele verden tolker informasjon, og returnerer resultater, sendt fra sentrale tjenere i USA. Andre eksempler på distribuert prosessering i global skala inkluderer kreft- og AIDS-forskning (for eksempel Screensaver Lifesaver-prosjektet[7] og FightAIDS@home[8] ved henholdsvis University of Oxford og Scripps Research Institute) hvor tusenvis av frivillige lar datamaskinene sine bidra til å løse problemer som ellers gjerne ville tatt mangfoldige år å løse, selv for superdatamaskiner. Disse prosjektene benytter alle Internett for å transportere pakker av informasjon mellom tjenermaskinene og de nærmest utallige nodene, hvor spesialtilpasset programvare tolker dataene.

Svakheter ved HTML

Nettverkteknologier muliggjør altså både World Wide Web og distribuert prosessering, og så lenge man kan kontrollere all informasjon - som i prosjektene nevnt ovenfor - kan man utnytte informasjonen til sitt fulle. I dag er dog svært mye informasjon nedfelt i HTML-dokumenter på WWW, samtidig som bruken av WWW er svært differensiert; bedrifter handler sammen (B2B-handel), privatpersoner leser nettaviser, kjøper aksjer, leter etter informasjon om det lokale legekontoret, og så videre. Det finnes med andre ord utallige arenaer som krever at datamaskiner kan utveksle ulike typer informasjon seg imellom, og store deler av den stadig voksende, og ukontrollerte, mengden av informasjon ekisterer i form av HTML-dokumenter på WWW.

Denne informasjonen gjøres tilgjengelig for menneskelige brukere via for eksempel nettlesere, men kan vanskelig brukes for andre formål. HTML, og også andre typer presentasjons- og prosedyreorienterte merkespråk, lar en nemlig utelukkende angi hvordan informasjonsfragmenter skal presenteres. Presentasjonssentrert merking angir parametre for layout og presentasjon, og kan dermed bistå mennesker (via brukeragenter som for eksempel nettlesere) i å fastslå mening i en layout-kontekst. Betydningen av informasjonsinnholdet er derimot ikke angitt, noe som gjør HTML-dokumenter ubrukelige for formål som krever ytterligere datamaskinell behandling. Alle ressurser som er tilgjengelige på WWW kan selvsagt hentes inn og til en viss grad tolkes av programmer, men å maskinelt fastslå betydningen til ulike biter av informasjon i HTML-dokumenter er en svært vanskelig prosess (Antoniou and van Harmelen 2004). Uten at en eksplisitt angir hvilken type informasjon en gitt ressurs inneholder, vil denne typen mening være relativt utilgjengelig for datamaskiner.

[7] `http://www.chem.ox.ac.uk/curecancer.html`
[8] `http://fightaidsathome.scripps.edu/`

Mens HTML fungerer utmerket for sitt tiltenkte formål, nemlig å gjøre spredte informasjonsressurser tilgjengelig for brukere i et distribuert nettverk (Berners-Lee 1991), er HTML som merkespråk altså ikke egnet til å bære informasjon om betydingen til informasjonsfragmenter i de ulike ressurser. HTML egner seg derfor svært dårlig for formål som krever distribuert prosessering, eller som grunnlag for for eksempel søk i informasjonsmengder (mer om dette i kapittel 3). For i større grad å kunne utnytte ressurser på WWW - for flere formål - må en altså kunne representere informasjonen på andre måter enn i form av HTML. Før oppgaven går i detalj omkring slike representasjonsformater, diskuteres liknende problemer knyttet til kunnskapsrepresentasjon innenfor kunstig intelligens (KI).

2.3 Kunstig intelligens (KI)

Ved fremveksten av digitale datamaskiner fra 1960-tallet av var det mange som så uante muligheter i datamaskinens kraft og virksomhetsområder. Innenfor 1960-1970-tallets KI-miljø var hovedmålsetningen å konstruere datamaskiner /
-programmer hvis "resonneringsevne" kunne sidestilles med, eller overgå, menneskets. Arbeidet innenfor KI konsentrerte seg om å gjøre maskiner i stand til å løse problemer som kan sies å kreve en eller annen form for intelligens[9]. For sjakkspilling antok en for eksempel at en menneskelig sjakkspiller for ethvert trekk tar utgangspunkt i samtlige mulige oppstillinger av brikker på sjakkbrettet, og leter seg frem til det best mulige trekket han kan utføre. Slike metoder, populært kalt prøving-og-feiling, ble implementert i programmer for å simulere kognitive prosesser, og ved hjelp av prøv-og-feil-søk kunne datamaskiner som var matet med informasjon om sjakkspillets regler og et gitt spills status konkurrere mot menneskelige utøvere. Et trekk foretatt av en datamaskin ble altså ikke valgt på grunnlag av tilfeldigheter, men etter nøye gjennomgang av en gitt situasjons muligheter. Etter hvert begynte sjakkspill som programmet Chess å slå menneskelige spillere - til og med "Grandmasters" (Copeland 1993). Enkelte KI-forskere hevdet derfor med utviklingen av slike programmer å ha tatt viktige steg mot simulert intelligens, men ikke alle var like optimistiske. KI-kritikeren Hubert Dreyfus påpekte blant annet i *What Computers Still Can't Do* (1979) at feltet ikke nærmet seg noen form for simulering av menneskelige tankeprosesser, blant annet fordi menneskets hjerne ikke jobber ut fra enkel prøving og feiling. En menneskelig sjakkmester vil eksempelvis ikke måtte gå stegvis

[9] I betydningen problemløsningsevne.

gjennom for eksempel 25.000 mulige trekk, men "automatisk" konsentrere seg om 100-200 muligheter (Dreyfus 1979).

2.3.1 Mikroverdener

Sjakkspilling var ikke det eneste området hvor en etter hvert lyktes i å la datamaskiner løse andre typer problemer enn matematiske beregninger på en tilfredsstillende måte. Et kjent eksempel på et program fra denne epoken er SHRDLU, et program for "understanding natural language" utviklet av Terry Winograd ved M.I.T (Winograd 1970). SHRDLU simulerer en robotarm som kan skille mellom og flytte på ulike typer virtuelle klosser, og er dessuten i stand til å holde en "dialog" (om objektene i den virtuelle verdenen) med menneskelige brukere. Man kan for eksempel be SHRDLU utføre kommandoer som "Pick up the biggest pyramid." (i), eller stille spørsmål som "What are you holding?" (ii) og "Is there anything which is bigger than every pyramid but is not as wide as the thing that supports it?" (Winograd 1970). Dersom spørsmålet (ii) gis etter at kommandoen (i) er utført, vil programmet svare med hvilken pyramide som "holdes opp" - for eksempel "The blue pyramid.". I tillegg til spørsmål og kommandoer kan en mate programmet med ny informasjon, gjennom input som for eksempel "I own the red pyramid." (iii).

SHRDLU ble av mange hevdet å være en suksesshistorie innen KI, og blant annet påstått å kunne forstå konsepter som for eksempel "own" (etter å ha blitt matet med informasjon á la (iii)). Dreyfus (1979) hevder derimot at programmer som SHRDLU ikke representerer kunstig intelligens, ettersom løsninger utviklet for et program som befatter seg med en avgrenset *mikroverden* ikke er generaliserbare. De prosedyrer som SHRDLU benytter seg av fungerer bare i et domene bestående av klosser som kan flyttes på. SHRDLU "forstår" dessuten bare et begrenset antall ord og setningstyper. Dersom SHRDLU for eksempel blir stilt et spørsmål utenfor sitt problemområde, vil det ikke kunne svare (Dreyfus 1979). Dreyfus påpeker videre at SHRDLU ikke har evne til å forstå begreper som "eie", etter å ha blitt matet med (iii), og at programmet bare handler ut fra en mengde predefinerte primitiver og relasjoner (Dreyfus 1979). SHRDLU relaterer bare deler av input til innebygde prosedyrer - det forstår ikke, og dersom programmet skal kunne "forstå" begreper som å 'eie' kan det ikke eksistere isolert av verden forøvrig. For å forstå hva det vil si å eie må en blant annet kunne fastslå hva det vil si å eie noe, samt at et dataprogram slett ikke kan eie noe som helst (Dreyfus 1979). De

ulike konsepter knyttet til begrepet 'eie' vil stadig vokse, noe som Dreyfus mener viser at SHRDLU egentlig ikke forstår hva det hevdes å forstå. Videre understreker han at at dataprogrammer som SHRDLU lykkes fordi en i ekspertsystemer har en begrenset mengde begreper som kan spesifiseres uttømmende (Dreyfus 1979). Kunnskap som av mennesker oppfattes som "common-sense" tas for eksempel gjerne for gitt, eller ses bort fra, i mikroverdener. Dataprogrammer som tillatter input fra brukere opererer dessuten gjerne ut fra antakelser med hensyn til tvetydighet i språk, og et program hvis problemområde er geometri vil gjerne tolke alle ord ut fra et matematisk perspektiv.

Systemer som befatter seg med mikroverdener kan med andre ord tilfredsstillende løse et konkret problem, men vanskelig generaliseres uten at en støter på problemer knyttet til for eksempel tvetydighet og ufullstendig informasjon. At dette er tilfellet ble i løpet av relativt kort tid bekreftet gjennom forsøk på å generalisere løsninger fra mikroverdener[10], og innenfor KI begynte en derfor etter hvert å konsentrere seg om problemer knyttet til representasjon av kunnskap (Dreyfus 1979).

2.3.2 Problem vs. metode

Programmet General Problem Solver (GPS), utviklet av Newell, Shaw og Simon var, i en begrenset grad, mer generaliserende enn andre programmer. SHRDLU kunne, som vi har sett, bare løse ett spesielt problem, mens GPS (og senere versjoner av programmet) lyktes i å løse forskjellige problemer. For eksempel kunne GPS løse både 'Misjonær og kannibal'- (MK) og 'Tårnet i Hanoi'-problemet. I 'Misjonær og kannibal'-problemet blir man presentert for en situasjon hvor det står 3 kannibaler og 3 misjonærer på bredden av en dyp elv som må krysses. For å krysse elven er bare 1 båt med plass til maksimum 2 personer tilgjengelig, og det kan aldri være et flertall av kannibaler på noen av elvebreddene; i et slikt tilfelle vil misjonærene bli spist opp, og programmet har feilet (Copeland 1993). 'Tårnet i Hanoi' utgjør i likhet med MK et logisk problem, men en forskjellig situasjon. I utgangspunket har man her tre trepåler, hvor det rundt den midterste er plassert ringer i ulik, og stigende, størrelse. Ringene skal så flyttes fra den midterste pålen til en av de andre pålene. Kun en ring kan flyttes per gang, og en stor ring kan ikke legges over en mindre ring (Copeland 1993).

På samme måte som SHRDLU, løste GPS problemer ved å transformere uttrykk fra en form (språk) til en annen form (formelle regler) (Sowa 2000). GPS lyktes dog i større

[10] Om generalisering skrev en av KIs "grunnleggere", John McCarthy, i 1987 blant annet "There simply is no most general context." (McCarthy 1987, s.1034).

grad å finne løsninger uavhengig av problemet, noe som gjenspeiles i at det klarte å løse begge de nevnte problemene. Dette var mulig fordi en for GPS hadde gjort et skille mellom problemdomenet og problemløsningsstrukturer for mål og delmål (McCarthy 1987). Selv om problemene som GPS kunne løse måtte ha en spesiell form og måtte kunne brytes ned og uttrykkes i form av en mengde tillatte regler - og derfor ikke var genereliserende som sådan, var det allikevel avgjørende at en med GPS begynte å skille mellom algoritme og problem (McCarthy 1987).

Til tross for at GPS ikke løste problemet med mikroverdener kan det derfor sies å ha lagt grunnlaget for kunnskapsrepresentasjon, hvor en representerer formalisert kunnskap i maskinleselige formater. Blant annet definerte Marvin Minsky, en av KIs pionérer, 'frames' som datastrukturer som representer stereotype situasjoner (Sowa 2000, s.144). Gjennom Minskys 'frames' kan kunnskap til en viss grad representeres i strukturerte enheter (Sowa 2000), og etter hvert ble det utviklet flere KR-språk med samme mål for øyet. For eksempel lar språk som Knowledge Interchange Format (KIF) en uttrykke "kunnskap" i et formelt språk for bruk av ulike systemer (Sowa 2000), og også programmeringsspråk som PROLOG (Programming for Logic) bygger på at ulike problemer løses av de samme algoritmene, og at "intelligensen" er nedfelt i datasettene (klausuler, regler, predikater, og så videre).

Mikroverdenen HTML

Med utviklingen av programmer som GPS begynte man altså å skille mellom uforanderlige algoritmer og selvbeskrivende datasett. En slik KR-formalisme gjenspeiles også i SGML, da enhver SGML-tolker er i stand til å tolke et hvilket som helst SGML-dokument. Steven R. Newcomb (2003) understreker viktigheten av dette ved å referere til hovedtanken bak SGML, slik den er forsøkt fortalt i en av Yuri Rubinskys[11] filmer:

> ... any information - *any* information - can be marked up in such a way as to be parsable (understandable, in a certain basic sense) by a single, standard piece of software, by any computer application, and even by human readers using their eyes and brains (Newcomb 2003, s. 31).

Denne separasjonen mellom algoritme og datasett i SGML ble videreført i Berners-Lees World Wide Web-prosjekt, ettersom HTML er en SGML-applikasjon. HTML inneholder grovt sett to typer informasjon; presentasjonsparametre i form av merker, og "innhold".

[11] Kjent talsmann for SGML.

Enhver algoritme for presentering av HTML kan presentere et hvilket som helst HTML-dokument, selv om de enkelte parametrene er forskjellige. HTML er med andre ord selvbeskrivende med hensyn til presentasjon. På grunn av at HTML-merker generelt sett ikke gir mening utover i en layout-kontekst[12] (ref. avsnitt 2.2.1), kan HTML derfor sies å representere en mikroverden knyttet til layout. Dermed oppstår det også et problem når det blir ønskelig å benytte informasjon fra HTML-dokumenter for andre formål enn det rent presentasjonsmessige, i og med at en simpelthen ikke har informasjon om andre begreper enn de knyttet til layout. I løpet av de siste årene har dette problemet blitt stadig mer gjeldende, ettersom svært mye tilgjengelig informasjon i dagens samfunn eksisterer i form av HTML-dokumenter. Samtidig som behovene for å benytte informasjon for andre formål enn det rent presentasjonsmessige er økende, er det dessuten ofte ønskelig å publisere ny informasjon i nettopp HTML, fordi WWW er en global Internett-applikasjon med millioner av brukere. Det har med andre ord oppstått et behov for å kunne representere informasjon om andre aspekter enn det rent presentasjonsmessige, også for allerede eksisterende ressurser på World Wide Web.

Gjennom blant annet nye kunnskapsrepresentasjonsspråk håper en å kunne eliminere noen av problemene knyttet til HTML og nevnte former for informasjons- / meningstap. En slik teknologi er World Wide Web Consortiums Web Ontology Language (OWL, diskutert i avsnitt 3.3). OWL bygger på formell-logikk i form av Description Logic (DL), mens DL på sin side er basert på Minskys frames. OWL representerer således en sammenheng mellom merkespråk og KI.

En annen 'KR-teknologi' for Web er emnekartstandardens XML Topic Maps, som er benyttet i det praktiske arbeidet med Hellerprosjektets vevapplikasjon (diskutert i kapittel 4). Før det gjøres rede for hvordan teknologier som OWL og emnekart kan bidra til å uttrykke informasjon som går tapt i HTML, gjøres det i kapittel 3 i detalj rede for de viktigste ingredienser og forutsetninger for en mer 'meningsfylt' verdensvev.

[12] Enkelte HTML-elementer, som for eksempel <address>, kan riktignok med rette sies å være meningsfulle med tanke på type merket informasjon, men dette gjelder bare en liten brøkdel av HTMLs merker.

Kapittel 3

Semantic Web

Ordet 'semantikk' kan oversettes med 'mening'; semantisk med 'meningsfylt'. 'Semantic Web' kan derfor oversettes med 'meningsfylt web', og er en betegnelse for en verdensvev hvor betydningen til informasjon nedfelt i ulike typer informasjonsressurser er eksplisitt angitt i maskinleselige formater. Semantic Web er altså en visjon om en vev hvor "meningen" i informasjonsressurser i større grad enn på dagens WWW er tilgjengelig for bruk av dataprogrammer.

3.1 Betydning på WWW

På dagens web er hovedparten av ressurser beskrevet i formater som gjør betydningen til elementene av informasjon nedfelt i dem lite tilgjengelig for datamaskiner. Et HTML-dokument inneholder som nevnt i oppgavens kapittel 2 bare informasjon om hvordan innholdet skal presenteres, ikke om betydningen av innholdet selv. Dersom en tekst i et HTML-dokument hevder at 'Helleren Skipshelleren ble brukt som boplass for steinalderfolk.' (Eksempel 3.1) vil de fleste mennesker sannsynligvis kunne slutte seg til at Skipshelleren er en heller, samt at det for flere tusen år siden levde mennesker (steinalderfolk) ved denne helleren. Et program i en datamaskin vil derimot ikke kunne identifisere betydningen til de ulike tekstfragmentene i et slikt HTML-dokument (med mindre det er spesialtilpasset). Fra Eksempel 3.1 vil et program ikke kunne avgjøre særlig annet enn at dokumentet inneholder tekst, hvilke tekstbiter som hører til hvilke overskrifttyper, hvilken tekstbit som hører til et gitt avsnitt, hvilken farge ulike tekstbiter skal vises i, og så videre. Informasjon om hva som beskrives vil for et dataprogram være

utilgjengelig. Semantic Web er derfor en betegnelse for en verdensvev hvor større deler av informasjonsressursene er deskriptivt merket på en måte som gjør informasjonen nedfelt i dem lettere prosesserbar for datamaskinell bruk *utover presenteringsformål* (Antoniou and van Harmelen 2004). På en semantisk verdensvev vil maskiner kunne lette trivielle oppgaver knyttet til informasjonsutveksling og -innhenting gjennom å tolke informasjonen nedfelt i de ulike informasjonsressurser.

HTML tillater riktignok inkludering av en begrenset mengde metainformasjon, som for eksempel nøkkelord og en beskrivelse av dokumentet, men denne informasjonen er lite strukturert og lite standardisert. HTML-metadata brukes dessuten ofte på måter som gjør det vanskelig å tillegge dem stor vekt. For eksempel vil en lett kunne inkludere uriktige / unøyaktige nøkkelord - det være seg bevisst inkludering av totalt irrelevante søkeord for manipulering av søkemotorer (noe en gjerne ser på 'lyssky', men høyt besøkte nettsteder), eller grunnet manglende kunnskap og valg av nøkkelord (for eksempel svært generelle nøkkelord), manglende enighet, og så videre. Grunnet manipulering og feilaktig bruk av nøkkelord tillegger dagens søkemotorer slike elementer liten vekt (Passin 2004). Metadata i HTML består videre stort sett av komma-separerte lister av verdier, hvor det ofte i prinsippet er umulig å si i hvilket forhold de ulike verdiene står til hverandre. Verdien av slike HTML-elementer er derfor relativt lav, semantisk sett.

```
<h1>Hellere</h1>
<p>Hellere kan ha vært brukt som boplasser for steinalderfolk.</p>
```

Eksempel 3.1: Utdrag fra tenkt HTML. I en nettleser vil 'Hellere' vises i overskriftsnivå 1, mens resterende tekst er merket som et avsnitt ("paragraph").

Problemet med prosedyre- og presentasjons-orientert merking - og behovet for mer semantisk merking - lar seg lett illustrere gjennom det dagligdagse problemet med å finne informasjon om et gitt tema på WWW ("Information Retrieval"). Ved leting etter informasjon på WWW vil en typisk benytte søkemotorer som Google eller Yahoo! for å forsøke å finne relevante dokumenter. Det tar relativt liten tid å taste inn en søkestreng og klikke på 'Søk' i Google, og et søk etter for eksempel 'Skipshelleren' vil mest sannsynlig returnere et begrenset antall treff i og med at en kan anta at det meste som er skrevet om Skipshelleren er norskspråklig. Man risikerer dog ofte å motta store mengder irrel-

evant informasjon, ettersom søkemotorer leter etter alle dokumenter på WWW (i deres indekser) som inneholder søkestrengen, eller deler av den, i en eller annen kontekst (kontekstuavhengig søk). Treffene rangeres så ut fra hvor mange forekomster av søkefrasen som finnes i de ulike dokumentene - gjerne kombinert med antall linker fra andre nettsteder til gitte informasjonsressurs (Google 2004). Trefflisten kan med andre ord fort vokse seg svært stor på en verdensomspennende vev. For eksempel returnerer et Google-søk etter 'Semantic Web' over 15 millioner treff (oktober 2005). Å finne akkurat hva man leter etter blant 15 millioner ressurser er naturligvis en lite tiltalende oppgave, og mest sannsynlig har man ikke tid til å undersøke mer enn et begrenset antall dokumenter. Selvsagt kan man være heldig og finne et relevant treff høyt oppe i trefflisten, men i en så overveldende informasjonsmengde kan man aldri vite om man valgte ut det mest relevante, eller beste treffet. Dette skyldes blant annet måten Google rangerer treffene sine på, men mest av alt skyldes det det faktum at alle nettsider (i Googles indekser) som inneholder frasen 'Semantic Web' - et eller annet sted i dokumentet - inkluderes i søkeresultatet. Det kan også tenkes at et søk til tross for gyldig og relevant søkefrase returnerer null treff, fordi søkemotoren for eksempel ikke kan identifisere synonyme ord i et dokument (Antoniou and van Harmelen 2004). Med en informasjonsmengde som er prosedyre-orientert merket er det sannsynligvis vanskelig å produsere særlig bedre søkeresultater enn hva søkemotorer som Google gjør, men dersom informasjonsressursene også var beskrevet i et merkespråk hvor betydningen til bestanddeler var eksplisitt angitt ved hjelp av 'semantiske' elementer, kunne datamaskiner gitt en bedre presisjon, samtidig som treffene kunne vært organisert etter helt andre metoder (betydning og identitet) enn hva som er mulig gjennom fulltekstsøk basert på form alene. I XTM-baser kan en for eksempel tillate fulltekstsøk hvor resultatene presenteres med utgangspunkt i emnetype, emnenavn og forekomster, eventuelt komplekse spørringer i form av tolog[1] eller TMQL[2]-uttrykk (Garshol 2004).

Grunnet Internett og WWWs størrelse og natur er det dog fremdeles usikkert i hvor stor grad problemene rundt informasjonsinnhenting ved hjelp av søk kan løses gjennom betydningsorientert merking alene. Ettersom det finnes millarder av informasjonsressurser på WWW vil en være avhengig av at eksisterende informasjonsressurser beskrives i metaspråk, at en kan stole på merkingen av informasjon, at klassifiseringer av ulike typer

[1] Spørrespråk for emnekart. Se Garshol (2002) for utfyllende informasjon.

[2] Topic Maps Query Language. Kommende standard-spørrespråk for emnekart. Se Garshol (2005) for en oversikt over TMQL.

informasjon er korrekt, samt at standardiserte metaspråk tas i utstrakt bruk (Passin 2004). Hvorvidt dette lar seg gjennomføre er uvisst, men for isolerte kunnskapsbaser (gjerne distribuerte, men tillitsbaserte) vil det utvilsomt kunne være fordelaktig å utnytte betydningsorienterte merkespråk, også med tanke på søk. En bedring av søk er dessuten i seg selv ikke et overordnet mål med en semantisk verdensvev, men en av flere muligheter som åpner seg når betydningen til informasjon angis eksplisitt i deskriptive og standardiserte merkespråk. Berners-Lee et al. (2001) beskriver i artikkelen *The Semantic Web* en visjon om hvordan en ved hjelp av intelligente agenter[3] på en semantisk vev kan automatisere daglige gjøremål som for eksempel bestilling av en legetime. Et dataprogram vil ikke kunne finne informasjon om kontortider ved å lese et tilfeldig valgt legekontors HTML-nettsider, men dersom de ulike strengenes betydning var eksplisitt angitt gjennom en markering av hvilke strenger som bærer informasjon om tidspunkt for kontortider og hvilke som angir ledige timer (hvor markeringens betydning var tilgjengelig for dataprogrammer) kunne en agent på egenhånd funnet frem til, og utvekslet informasjon med andre agenter (for eksempel legekontorets) om åpningstider og ledige timer. Deretter kunne agenten sjekket andre avtaler - lagret i en digital almanakk, presentert brukeren for mulige timer, og gjennomført bestillingen. Andre scenarier finner en hos blant annet Antoniou og van Harmelen (2004), Passin (2004) og Ding et al. (2002). Felles for dem alle er at enkle oppgaver som på dagens verdensvev må gjøres for hånd av menneskelige brukere på en semantisk verdensvev kan utføres av digitale agenter / datamaskiner, fordi informasjon vil være lagret på en måte som gjør at også datamaskiner kan identifisere betydningen til, og dermed utnytte, ulike informasjonsfragmenter.

Det kan riktignok hevdes at maskiner allerede i dag automatisk nyttiggjør seg informasjon på WWW, og at det derfor allerede eksisterer en semantisk verdensvev. For eksempel samler såkalte Shopping Agenter inn informasjon om ulike produkter fra forskjellige produktleverandørers nettsider. På www.kelkoo.no kan en for eksempel søke etter og sammenligne priser blant et utvalg produktleverandører. Dette gjøres ved å tolke HTML-koden fra de ulike produktleverandørers nettsider gjennom forhåndsprogrammerte slutninger om hvilke tekstbiter i HTML-dokumentene som angir pris, hvilke som angir produktnavn, produktbeskrivelse og så videre. Disse informasjonsbitene trekkes ut fra HTML-koden og fremstilles for brukeren. Kelkoo.no kan for eksempel bistå en bruker i å finne ut i hvilken butikk en gitt type mobiltelefon synes å være billigst. Dersom en pro-

[3]Intelligente agenter er dataprogrammer som for eksempel kan trekke slutninger på egenhånd, og slik også finne implisitt informasjon. Se Wooldridge (2002) for utfyllende informasjon.

duktleverandør endrer vesentlig på sine nettsiders HTML-kode vil en dog måtte endre på koden i selve shoppingagenten. 'Semantikken' ligger nemlig hardkodet i selve programmet som tråler nettsteder på jakt etter informasjon, ikke i HTML-koden. En (mer) semantisk verdensvev innebærer at en flytter semantikk fra applikasjonen til dokumentet (Uschold 2001) (ref. avsnitt 2.3.2). En 'semantisk agent' vil i motsetning til Kelkoo ikke være spesiallaget for ett formål, men kunne utføre et uttall forskjellige oppgaver ved å tolke ulike informasjonsressurser - så lenge disse er beskrevet i et 'semantisk språk' hvor elementenes betydning er angitt. På en semantisk verdensvev vil det altså være meta-språkenes semantikk og regler som vil være hardkodet i agentene (Uschold 2001). For å nærme seg visjonen om en semantisk vev kreves med andre ord mer betydningsorienterte merkespråk.

3.1.1 Betydningsorientert merking

HTML var i utgangspunktet ikke ment å være et merkespråk med støtte for utallige[4] typografiske virkemidler, men da det er et tekst- og presentasjonsorientert merkespråk er det som nevnt ikke anvendelig for særlig annet enn presentasjon av data for menneskelige brukere. Dersom det skal være mulig å dele og gjenbruke data mellom ulike applikasjoner (W3C 2005) kreves det andre typer merkespråk, med mulighet for vektlegging av elementenes betydning. Med bakgrunn i dette, samt arbeidet med et stilspråk (Cascading Style Sheet (CSS)) for enklere manipulering av stil over en mengde HTML-dokumenter samtidig, ble eXtensible Markup Language (XML) utviklet (Cagle 2000).

eXtensible Markup Language

I likhet med HTML bygger XML på SGML, og også XML er enklere enn SGML. Mens HTML begrenser seg til en mengde predefinerte elementer, kan en dog gjennom XML angi egendefinerte elementer. XML er nemlig en standard for hvordan en kan beskrive applikasjons-spesifikke merkespråk (egne sett med elementer og attributter); XML-vokabularer (W3C 2004a).

Alle XML-dokumenter må oppfylle krav til knyttet til velformethet - blant annet må det bare eksistere ett rot-element i et gitt XML-dokument, alle elementer må lukkes, det tillates ikke overlapping av elementer (tillatt i HTML), og så videre (W3C 2004a).

[4]Mye ble i WWWs barndom presset frem gjennom nettleser-'kriger' hvor det gjaldt å støtte så mange, og gjerne fancy (<blink>,<marquee>, etc.), elementer som mulig.

Grunnet slike restriksjoner vil alle velformete XML-dokumenter ha en slags trestruktur hvor elementene er nøstet nedover i treet, og hvor rekkefølgen til elementene er vesentlig. XML-dokumenter kalles videre gyldige dersom de tilfredsstiller krav nedfelt i Document Type Definition (DTD) eller XML Schema (W3C 2004d). DTDer og XML Schema benyttes for å beskrive XML-vokabularer og tillater en blant annet å spesifisere lovlig rekkefølge på elementer, tillatte attributter for gitte elementer, tillatte verdier for elementer og attributter, med mer. Velformethet angir med andre ord syntaktiske kriterier som gjelder for alle XML-dokumenter, mens gyldighet angir forholdet mellom et sett av dokumenter og deres semantiske regler. Fordi alle XML-dokumenter som SGML-dokumenter har samme oppbygning vil enhver standardisert XML-parser (tolker) kunne lese ethvert XML-vokabular (ref. avsnitt 2.3.2), og dermed inneholde en strukturert representasjon av informasjonen i dokumentet.

Med hensyn til en betydningsorientert vev er ikke det viktigste aspektet ved XML at alle XML-dokumenter har en trestruktur, eller at man kan angi datatyper i XML Schema (selv om også disse sidene ved XML er viktige), men at man ved hjelp av egne XML-vokabularer kan skape strukturerte dokumenter - i et utbredt og standardisert format - som bærer andre typer metainformasjon enn det rent presentasjonsorienterte[5]. Eksempel 3.2 viser en mer deskriptiv versjon av Eksempel 3.1, hvor betydningen til de forskjellige informasjonsfragmentene indikeres i merkenavnene. Ut fra elementene i et slikt XML-dokument kan en i høyere grad avgjøre hvilken type informasjon det er snakk om, enn hva som er mulig ut fra HTML-versjonen i Eksempel 3.1. Dersom et dataprogram er kjent med betydningen av de ulike elementene i et slikt XML-dokument, vil det (lettere) kunne trekke ut og benytte de ulike informasjonsfragmenter fra dokumentet. Alle former for slutninger må dog fremdeles kodes inn i applikasjonen, hvis kode må være tilpasset hvert XML-vokabular. Semantikken befinner seg med andre ord også her i applikasjonen, men deler av den kan være overført til dokumentet i det et menneske (programmerer) lettere kan slutte seg til hva som beskrives.

[5]XHTML er en 'utgave' av HTML som konformerer med XML (XHTML-dokumenter kan derfor tolkes av XML-parsere), uten at dette gjør XHTML til et 'semantisk' merkespråk.

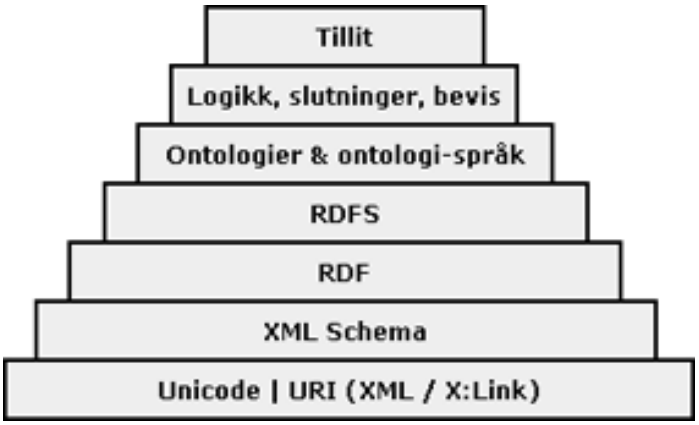
```
<?xml version='1' encoding='utf-8' ?>
<historisk_boplass type='heller'>
    <navn>Skipshelleren</navn>
    <benyttet_av>Steinalderfolk</benyttet_av>
</historisk_boplass>
```

Eksempel 3.2: Mulig XML-versjon av Eksempel 3.1. Elementene og attributtet ('type' i elementet 'historisk_boplass') er her egendefinerte ut fra et tenkt behov for å representere informasjon om hellere i et deskriptivt merkespråk.

Det finnes i dag flere forsøk på å utvikle XML-baserte semantiske netteknologier, og World Wide Web Consortiums (W3C) viktigste 'standarder'[6] for Semantic Web er Resource Description Framework (RDF) og Web Ontology Language (OWL), hvor OWL bygger på RDF. Gjennom disse nye språkene søker en å nærme seg en mer semantisk verdensvev. Blant annet kan nyttigjøring av strukturert metainformasjon tillate intelligente agenter å utføre oppgaver som på dagens WWW krever menneskelig innsats. Figur 3.1 viser en forenklet versjon av Berners-Lees 'Semantic Web "Layer Cake"' (Berners-Lee 2004), som angir ulike komponenter som kreves for den typen semantisk vev som W3C ser for seg. I bunn finner vi 'adresserings'- og XML-teknologier som er viktige for at datamaskiner skal kunne lokalisere og ekstrahere ulike typer informasjon som det refereres til. Deretter følger RDF/RDF Schema og OWL som lar en beskrive ressurser og ontologier. På topp står logikk og tillit ('trust'). Logikk kreves for at agenter skal kunne trekke slutninger fra foreliggende datasett, samt for å kunne bevise at en fattet slutning er korrekt.

Dersom en skal kunne stole på informasjon, særlig dersom agenter skal kunne utveksle sensitiv informasjon, behøves en form for tillit. Slik tillit kan formaliseres og etableres gjennom bruk av digitale signaturer og sertifiserte tredjeparter (Antoniou and van Harmelen 2004). Digitale signaturer muliggjør nemlig identifisering av digital informasjon gjennom digitale "fingeravtrykk". Ved hjelp av digitale signaturer kan en altså fastslå hvorvidt noe er sendt fra en gitt avsender, eller ikke. Sertifiserte tredjeparter er uavhengige bedrifter eller organisasjoner (tredjeparter) som opptrer som for eksempel godkjente (underlagt offentlig eller privat kontroll) utstedere av digitale signaturer, eller som på andre måter "går god for" bedrifter, organisasjoner eller privatpersoner (Passin 2004).

[6] W3C publiserer ikke standarder, men anbefalinger; 'recommendations'.



Figur 3.1: Forenklet versjon av Berners-Lees Semantic Web "Layer Cake".

De to bestanddelene av tillit er med andre ord viktige elementer for å kunne ansvarliggjøre avsendere på både WWW og en eventuell semantisk vev, og uten tillit kan ikke potensialet i en betydningsorientert vev fullt ut realiseres (Antoniou and van Harmelen 2004). Det forskes derfor i dag aktivt på dette området.

Semantiske netteknologier

En semantisk verdensvev er altså avhengig av standardiserte merkespråk som kan representere mening knyttet til typer innhold. For å kunne la et dokument bli selvbeskrivende med hensyn til Eksempel 3.2, må en for eksempel kunne uttrykke at en 'heller' er en type 'historisk boplass'. Videre må en kunne uttrykke i hvilket forhold ulike typer står til hverandre, samt entydig kunne identifisere typer og ressurser. Utvetydige identitetskriterier er nødvendige for å kunne skille mellom ulike informasjonsenheter og -ressurser, mens å kunne uttrykke relasjoner er nødvendig for å blant annet kunne si at en 'heller *ble benyttet av* steinalderfolk'.

For denne oppgavens del defineres en 'semantiske netteknologi' som en 'information overlay'-teknologi som evner å uttrykke:

- utvetydige identitetskriterier
- klasser
- objekter
- relasjoner

'Information overlay' betyr her at en semantisk netteknologi er en metateknologi som beskriver eksisterende ressurser, uavhengig av ressursenes format. Formålet til semantiske netteknologier er å tilføre WWW et lag av metainformasjon knyttet til meningsinnholdet i eksisterende ressurser, utover i en layout-kontekst (ref. avsnitt 2.2.1). Slike teknologier vil gjennom å være basert på ovenfornevnte kriterier kunne uttrykke entydige påstander om eksisterende ressurser, slik at informasjon fra ulike ressurser i større grad enn på dagens WWW kan gjøres tilgjengelig for datamaskinell prosessering. Semantiske netteknologier tilfører altså WWW et 'abstrakt' lag av metainformasjon.

Ettersom ontologier spiller en vesentlig rolle i arbeidet med å modellere og gjøre informasjon tilgjengelig for datamaskinell prosessering på en semantisk verdensvev, og også er et sentralt begrep innenfor emnekart og XTM, vil oppgaven gå i detalj omkring ontologi og ontologiutvikling i avsnittet under. Det vil også gjøres rede for enkel logikk og grunnleggende mengdelære, ettersom dette ligger til grunn for resonnering over informasjon i ontologier og dermed også for XTM, modellering av emnekart-programvare og vevapplikasjonen utviklet som en del av denne mastergradsoppgaven. Problemstillinger rundt tillit og bevis går oppgaven derimot ikke inn på, men henviser den interesserte leser til Berners-Lee et al. (2001); Passin (2004); Antoniou og van Harmelen (2004).

3.2 Ontologier

'Ontologi' er et begrep som går igjen i litteratur på områder som for eksempel Knowledge Representation (KR), Semantic Web og emnekart. Begrepet er hentet fra filosofi og omtaler her studien av det værende; alle typer entiteter - konkrete og abstrakte - som verden består av (Sowa 2000). Innenfor de ovenfornevnte områdene har begrepet fått en litt annen betydning, og betegner her de 'tingene' som det kan snakkes om i et system (Passin 2004). En ofte sitert (for eksempel Ding et al. (2002); Antoniou og van Harmelen (2004)) og mer formell definisjon finner vi hos Gruber (1993);

> An ontology is an explicit specification of a conceptualization (Gruber 1993, s.199).

'Specification of a conceptualization' refererer her til en modellering av alle objekter - og relasjoner mellom dem - innenfor et diskursdomene[7], mens eksplisitt betyr at typene til objektene og begrensninger hva gjelder domenekardinalitet, etc. er eksplisitt angitt (Ding

[7]Den delen av verden som beskrives.

and Foo 2002) (altså befinner de seg ikke bare i tankene til designeren). For kunnskapsbaserte systemer er det værende lik det som er representert, og man kan ikke anta at det eksisterer objekter eller konsepter utenom de som er beskrevet i en ontologi. Gruber fremholder dessuten at en ontologi skal være formell - maskinleselig (Gruber 1993). Det vil si at i en ontologi er alle 'ting' i diskursdomenet modellert og eksplisitt angitt i et maskinleselig format. Formålet med ontologier er å forsøke å gjøre den modellering og tolkning en har av gitt domene eksplisitt, for å muliggjøre gjenbruk og utveksling av informasjon mellom ulike systemer. Gjennom ontologier kan en altså gjøre informasjon om ens forståelse av et diskursdomene tilgjengelig for andre mennesker eller dataprogrammer (Gruber 1993).

Ettersom 'Ontologi' er et begrep som de siste årene har blitt tatt i bruk innenfor mange ulike miljøer - for eksempel KR, kunstig intelligens (AI), etc. - vektlegger Guarino (1998) at det eksisterer et skille mellom konseptualisering og ontologi. Konseptualisering angir det filosofiske aspektet av en ontologi; to ontologier kan være definerte i ulike vokabularer, men dele en felles konseptualisering (Guarino 1998). Guarino påpeker videre at en ontologi bare indirekte kan spesifisere en konseptualisering. Dette begrunnes gjennom en formell ('matematisk') analyse av hva henholdsvis en konseptualisering og en ontologi er, noe som munner ut i en omarbeidet definisjon av 'Ontologi' slik begrepet er definert av Gruber. Han understreker deretter at en ontologi er språkavhengig, mens en konseptualisering er språkuavhengig. Innenfor blant annet AI har de ulike termene blitt brukt om hverandre, men en separasjon er viktig dersom en skal kunne snakke om for eksempel ontologi-deling/-utveksling, eller slå sammen to ontologier til én ny ontologi (Guarino 1998). Denne oppgaven vil i det følgende forholde seg til Guarinos terminologi, og benytter begrepet 'ontologi' om den språkavhengige modelleringen av en konseptualisering, for eksempel en XTM-ontologi.

Ontologier deler altså diskursdomener inn i undermengder som representerer vesentlige egenskaper ved objektene de inneholder. Undermengdene i et diskursdomene representeres med andre ord type eller klasse, og medlemskap i klassen er det samme som predikering (Hellere = $\{x$: x er et overheng av berg$\}$). Ontologier kan altså brukes for å eksplisitt uttrykke ulike *typer* av informasjon (objekter i diskursdomenet) og relasjoner (forskjeller i egenskaper) typer imellom. Gitt at en ontologi på en betydningsorientert vev foreligger i et maskinleselig format med utvetydige identitetskriterier for hver enkelt type, vil et dataprogram kunne fastslå betydningen til et objekt ved å identifisere objektet og dets egenskaper i den anvendte ontologi. At det vi beskriver i ontologier angis eksplisitt

tillater med andre ord at dataprogrammer kan designes for å resonnere over informasjon nedfelt i filer og dokumenter, og nettopp derfor vil ontologier utgjøre en viktig del av en semantisk vev. Ved hjelp av ontologier kan et dataprogram fastslå hvilken type et gitt objekt tilhører, og ikke tilhører, samt hvordan det gitte objektet forholder seg til andre objekter. For eksempel kan en ved hjelp av en arkeologi-ontologi beskrevet i et emnekart avgjøre hvorvidt to emner er av samme type - for eksempel 'heller', hvordan de er relatert til hverandre - for eksempel 'geografisk nær', og så videre. Informasjon som tidligere ville vært hardkodet i dataprogrammer (ref. kapittel 2) kan altså beskrives i ontologier, og gjennom formelle ontologispråk gjøres tilgjengelig for bruk på tvers av nettverk og applikasjoner.

3.2.1 Resonnering over ontologiske kategorier

Det følger av det som allerede er sagt at en ontologi inneholder en mengde mengder. For en informasjonsmodell som er beskrevet i en ontologi kan vi derfor benytte mengdelære og -operasjoner som et redskap for å resonnere over informasjonsmodellen. Det gjøres her kort rede for de viktigste mengdeoperasjonene som ligger til grunn for programvare og vevapplikasjon utviklet for Hellerprosjektet. For eksemplenes del gjelder S = $\{x$: x er av stein$\}$, T = $\{x$: x er av tre$\}$, BE = $\{x$: x er av bein$\}$, R = $\{x$: x er et redskap$\}$, F = $\{x$: x er et funn$\}$.

Tom og universelle mengde

Mengden som ikke inneholder noen elementer kalles den tomme mengde og angis ved $\emptyset$. I motsatt ende finner vi den universelle mengden U (eng. "universal set"), som inneholder alle objekter i et diskursdomene. Rektanglene i Figur 3.5 (side 42) utgjør U for de respektive mengdene. U vil alltid inneholde $\emptyset$, siden $\emptyset$ er en submengde av enhver mengde; $\emptyset \subset$ A for alle A (Halmos 1998).

Union

Unionen av to mengder A og B ("A union B") er mengden A $\cup$ B = $\{$x: x $\in$ A *eller* x $\in$ B$\}$ (Haggarty 2002). Unionen av to mengder er altså en mengde som inneholder samtlige elementer i A, eller B, eller i begge. I Figur 3.5 tilsvarer dette alle elementer i (iii). Eksempler: Mengden av alle objekter som enten er tre, eller bein: $TB = T \cup BE$. Alle materialer (stein, tre, eller bein): $M = S \cup T \cup BE$.

Snitt

Snittet av to mengder A og B ("A snitt B") er mengden A ∩ B = {x: x ∈ A *og* x ∈ B} (Haggarty 2002). Snitt betegner med andre ord de elementer som finnes i både A og B ('samtidig'). I Figur 3.5 tilsvarer dette A i (ii). Eksempler: Mengden av stein-redskaper: $SR = R \cap S$. Alle funn av bein eller tre: $BF = F \cap (BE \cup T)$.

Komplement

Komplementet av en mengde A med hensyn på en annen mengde B er mengden A - B = {x: x ∈ A *og* x ∉ B} ("A ikke-B") (Haggarty 2002). I Figur 3.5 (iv) er -B lik alle elementene i A (elementene 'utenfor' B). I Figur 3.5 (i) er derimot alle elementene i A også i B, og derfor er A - B her lik ∅. Eksempler: Alle redskaper som ikke er tre: $RIT = R - T$. Alle funn av stein og bein som ikke er redskaper: $FBT = (F \cap (S \cap BE)) - R$.

Sekvenser og kryssprodukt

Den innbyrdes rekkefølgen til elementene i en mengde er vilkårlig. For to mengder A og B eksisterer det alltid en tredje mengde C = {A, B}, hvis elementer er mengdene A og B. Mengden C kalles her et uordnet par. I uordnede par er den innbyrdes rekkefølgen til elementene vilkårlig (Halmos 1998). For å kunne snakke om bestemte egenskaper mellom bestemte elementer i to eller flere mengder må en derimot kunne omtale ordnede par (sekvenser) og mengder av ordnede par, fordi en for en relasjon mellom elementer i to mengder må vite hvilke av elementene som er 'forbundet'. Det ordnede paret av elementene a og b, med første 'koordinat' a og andre 'koordinat' b, er mengden {a, b} definert ved <a, b> = {{A}, {A, B}} (Halmos 1998, s. 23). I ordnede par er den innbyrdes rekkefølgen til elementene vesentlig. Sekvensen <1,2,3> regnes her som forskjellig fra sekvensen <1,3,2>. For et vilkårlig antall elementer n i en sekvens S kalles S en n-tuppel. Mengden som består av alle ordnede par av A og B kalles *kryssproduktet* (Cartesisk produkt) av A og B, og betegnes med A × B.

Relasjoner

En relasjon representerer en sammenheng mellom elementer i mengder. En relasjon med bare ett argument (element) kalles gjerne en 1-plass relasjon, og angir en egenskap ved et objekt (eksempel: $Heller(x)$) (Moe 2003). Et eksempel på en 2-plass, eller *binær* relasjon,

er forelder-barn relasjonen - forelder(X, Y) - som gjelder mellom elementene X og Y dersom X er en forelder av Y. Formelt sett er en binær relasjon:

> A set R is a binary relation if all elements of R are ordered pairs, i.e., if for any z ∈ R there exist x and y such that $z = (x, y)$ (Jech 2002).

Hvis R er en binær relasjon mellom mengder, så gjelder xRy dersom (x, y) $\in R$. Hvis A er en mengde og R er en relasjon på A, så sies R å være refleksiv dersom xRx for alle $x \in A$. Dersom $xRy \Rightarrow yRx$ for alle x og y i A kalles R symmetrisk, hvis (xRy og $yRx \Rightarrow x = y$) for alle x og y i A så er R antisymmetrisk, mens R er transitiv dersom (xRy og $yRz \Rightarrow xRz$) for alle x, y og z i A (Haggarty 2002, s.61).

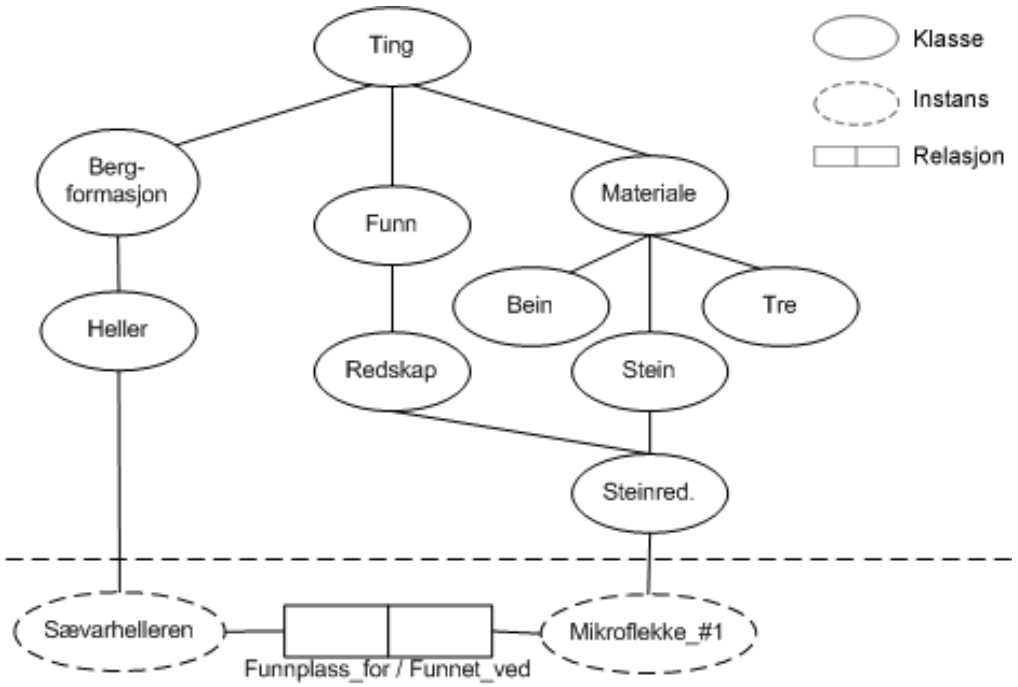
I prinsippet er det ingen øvre grense for hvor mange ulike roller / argumenter som kan delta i en relasjon, og i n-ære relasjoner kalles n relasjonens aritet (Moe 2003). Relasjoner med aritet 3 kalles tripler. I en relasjon $forelder(X, Y)$ hvor Xs rolle er forelder og Ys rolle er barn angir X relasjonens *domene* (eng. "domain"), mens Y angir relasjonenes *rekkevidde* (eng. "range"). En relasjon $R = Funnet\_ved(Funn, Heller)$, hvor første element angir relasjonens domene og andre element angir dens rekkevidde, gjelder altså dersom $Funn$ ble $Funnet\_ved\ Heller$. Vi ser videre at ettersom vi kan angi at $NyRel = (R, Y)$, som igjen kan benyttes i atter nye relasjoner, så er i prinsippet binære relasjoner alt vi trenger for å kunne uttrykke n-ære relasjoner hvor $n > 1$.

3.2.2 Ontologiutvikling

Ontologier i større systemer og/eller myntet på utveksling av informasjon bør utvikles systematisk for blant annet å hindre definering av uriktige typer, eller overlapping, fordi feilaktige ontologier - for eksempel selvmotsigende ontologier - kan resultere i feilaktige slutninger og andre typer feil. 'Ontologiutvikling' (eng. "Ontology Development", "Ontology Engineering") betegner prosessen hvor en identifiserer og beskriver klasser og egenskaper ved klasser, samt instanser av klasser i diskursdomenet - en prosess som minner om Entity-Relationship- (ER), Enhanced Entity-Relationship[8] (EER) -modellering og objekt-orientert design. Et eksempel på en ontologi over hellere på Vestlandet bør for eksempel inneholde en eksplisitt angivelse av egenskaper ved hellere generelt, og de utvalgte hellerne på Vestlandet spesielt (instanser av konseptet 'heller'). Samtidig bør en unngå at konsepter - for eksempel 'flint' - defineres flere steder i ontologien - for eksempel både som en type 'stein' og som en type 'redskap' (utenom som en følge av arv).

[8]Se Elmasri og Navathe (2004) for utfyllende informasjon om ER- og EER-modellering.

Egenskapene ved konseptet 'heller' (for eksempel hva en heller er) hører her til konseptualiseringen - de er de samme uavhengig av språk, men ontologien må også representeres i et språk for bruk av datamaskiner (for eksempel XTM eller OWL). Figur 3.2 viser et utdrag av en tenkt ontologi over arkeologiske funn i hellere på Vestlandet. I denne ontologien er alle entiteter i diskursdomenet modellert som en eller annen type 'Ting'. Videre er 'Redskap' modellert som en type 'Funn'. For enkelte formål vil dette kanskje være en grei løsning, men det kan tenkes at en egentlig ikke ønsker å angi alle former for redskaper som en type 'Funn' - for eksempel om en på et senere tidspunkt ønsker å beskrive redskaper som ikke også er arkeologiske funn. Hvorvidt steinredskaper bør modelleres som en type stein kan sikkert også diskuteres, men avhenger av ontologiens formål. Ettersom det lett kan oppstå feil i ontologier som ikke er utviklet gjennom en form for systematisk prosess, eller er særlig gjennomtenkte, vil det derfor svært ofte være hensiktsmessig å foreta en systematisk ontologiutvikling.



Figur 3.2: Utdrag av tenkt ontologi over hellere og arkeologiske funn.

Noy og McGuinness gir i *Ontology Development 101: A guide to creating your first ontology* (2001) en steg-for-steg innføring i ontologiutvikling. Første steg i denne guiden er å presist definere diskursdomenet - for hvilken del av verden ontologien skal gjelde,

og hvilken type informasjon ontologien skal inneholde. Disse aspektene ved en ontologi vil være avhengig av ontologiens bruksområder, både med tanke på domene og typen informasjon som modelleres (for eksempel med tanke på detaljnivå). Det finnes ikke noen fasit for hvordan en gitt ontologi skal være, så lenge den fyller sin funksjon og relevante spørsmål innenfor diskursdomenet kan besvares med utgangspunkt i den. For eksempel vil spørsmålet 'Hva er en flintkniv laget av?' være relevant for en ontologi over arkeologiske funn. Ettersom en av hovedtankene bak maskinleselige ontologier er utveksling og gjenbruk av informasjon, kan det være en fordel å undersøke mulighetene for å inkludere, eller bygge på eksisterende ontologier. Dersom allerede eksisterende ontologier er tilgjengelige vil dette både kunne redusere tidsbruken for ens egen ontologiutvikling, samt kunne bidra til å gjøre resulterende ontologi bedre egnet for utveksling og gjenbruk (Noy and McGuinness 2001). For noen områder vil en dog måtte utvikle egne ontologier, noe som blant annet har vært tilfellet for Hellerprosjektet.

Etter at innledende undersøkelser er foretatt vil neste steg være å identifisere viktige begrep i ontologien, samt definere klasser og et klasse-hierarki (taksonomi). Det finnes her to hovedinnfallsvinkler; top-down og bottom-up (Noy and McGuinness 2001). Ved top-down beveger en seg fra det mest generelle i ontologien, til det mest spesielle. For eksempel kan en si at alle objekter i en hellerontologi har en felles overtype: 'ting'. En kan så identifisere og dokumentere undertyper ved å bevege seg fra det mest generelle objektet i ontologien ('ting') til det mest spesielle objektet. Bottom-up følger samme fremgangsmåte, bortsett fra at en her først identifiserer det mest spesielle objektet og beveger seg fra dette objektet til det mest generelle objektet. De to fremgangsmåtene kan selvsagt kombineres underveis i arbeidet, og hvilken som vektlegges vil ofte avgjøres av faktorer som personlige preferanser eller identifiserte utgangspunkt.

Når en har fastslått ontologiens klasser og stilt dem opp i et hierarki kan en definere egenskaper ved klassene. For eksempel vil ethvert 'funn' i en arkeologi-ontologi ha et navn eller identifikasjonsnummer, være funnet ved et bestemt funnsted, og så videre. En kan også angi tillatte verdityper (heltall, tekststreng, etc.) og antallet tillatte verdier for egenskaper (for eksempel bare ett identifikasjonsnummer), med det formål å begrense mulige feilkilder eller forvirring. Etter at klasser og egenskaper ved klasser er identifisert og definert, kan en så opprette individuelle instanser av de ulike klassene. I Figur 3.2 er eksempelvis Sævarhelleren angitt som en instans av klassen 'heller', og innehar derfor alle egenskaper som enhver 'heller' har. De spesifikke verdiene for Sævarhellerens egenskaper vil dog skille seg fra verdiene for andre hellere, for eksempel er navnet ulikt

Hallgrimshellerens.

Fremgangsmåten som er beskrevet av Noy & McGuinnes (2001), og i korte trekk gjengitt ovenfor, er dog ikke den eneste muligheten man har ved ontologiutvikling. Man kan for eksempel benytte, eller kombinere nevnte metode med Formal Concept Analysis (FCA) (Wolff 1993), noe som er forsøkt gjort under utviklingen av Hellerprosjektets ontologi. Dersom en benytter FCA må en også utføre enkelte av stegene som er beskrevet ovenfor - for eksempel identifisere ulike klasser og egenskaper ved klassene, men FCA tilbyr en mer matematisk metode for å fylle inn verdier for instanser. FCA ordner dessuten klassene mer systematisk i et nettverk (eng. "lattice") ut fra antallet instanser av de enkelte klasser, ikke i et taksonomisk hierarki basert på typer, enn hva som mest sannsynlig vil være tilfellet ved en ren manuell ontologiutvikling slik prosessen er beskrevet i Noy og McGuinness (2001).

Uansett fremgangsmåte vil resultatet av en ontologiutviklingsprosess være en formalisert beskrivelse av et spesifikt domene. Ontologier utviklet for spesielle program / -miljø kalles derfor gjerne begrensede ontologier ("limited ontologies") eller mikroverdener (Sowa 2000). For eksempel vil ikke Hellerprosjektet ha som ambisjon å representere all informasjon i verden i sin ontologi, men forsøke å representere informasjon om ulike hellere, funnkategorier, spesifikke funn, og så videre - på et, for Hellerprosjektet, hensiktsmessig detaljnivå. På denne måten blir ontologien nyttig for sitt tiltenkte formål, samtidig som detaljer som gjerne kan være ønskelige i andre applikasjoner vil falle bort. At ontologier tilpasses den enkelte applikasjon og at ulike applikasjoner gjerne representerer de samme objektene på ulike måter bidrar til at kunnskapsdeling, -utveksling og -gjenbruk kan bli vanskelig, om ikke umulig (Sowa 2000), på samme måte som en innenfor kunstig intelligens feilet ved generalisering av mikroverdener (ref. avsnitt 2.3.1).

3.2.3 Organisering av kunnskap i ontologier

Taksonomier

Aritosteles' logiske undersøkelser av kategorier, forholdet mellom dem og sannhetsbevarende slutninger kan sies å ligge til grunn for ordning av ontologisk kunnskap i taksonomier. Taksonomier er strukturer som inneholder informasjon om foreldre-barn relasjoner (superklasse-subklasse) mellom objekter i ontologier. Aristoteles identifiserte i sine undersøkelser fire grunntyper kategoriske påstander:

Alle A er B.
Noen A er B.
Ingen A er B.
Noen A er ikke B.

A og B er her variabler som angir en hvilken som helst mengde elementer. Disse påstandstypene ligger til grunn for syllogismer - logiske slutninger (argument) som består av tre deler; to premiss og en konklusjon (Sowa 2000). Et eksempel på en enkel syllogisme om hellere kan være: "Fordi alle overheng av berg beskytter mot fuktighet (i) og fordi alle hellere er overheng av berg (ii), så beskytter alle hellere mot fuktighet (iii)". I dette eksempelet er (i) og (ii) premissene som leder til konklusjonen (iii). Gjennom bruk av variabler presenterte Aristoteles fire hovedtyper generaliserbare syllogismer (Figur 3.3) (Sowa 2000).

Barbara:	*Celarent:*
(i) Alle A er B	(i) Alle A er B
(i) Alle B er C	(i) Ingen B er C
(ii) Alle A er C	(ii) Ingen A er C
Darii:	*Ferio:*
(i) Alle A er B	(i) Noen A er B
(i) Noen B er C	(i) Ingen B er C
(ii) Noen A er C	(ii) Noen A er ikke C

Figur 3.3: Aristoteles' syllogismer (Sowa 2001). Navnene i kursiv angir huskeregler for oppbygningen av syllogismene, tatt i bruk av middelalderens filosofer.

Slutningsmønstre Barbara og Darii tillater nedarving av egenskaper fra henholdsvis superklasse til subklasse og fra klasse til instans, i moderne programmeringsspråk (Sowa 2000). Ved subtyping arves egenskaper fra en type konsepter ('ting') i en taksonomi til denne typens subtyper og/eller instanser. For eksempel er 'Bein' modellert som en type 'Materiale' i Figur 3.2, og innehar - arver - derfor alle egenskapene som er avgjørende for å definere en klasse som et materiale (Barbara), mens en konkret instans av klassen 'Bein' vil arve egenskaper fra klassen 'Bein' (Darii). Slutningsmønstrene Ferio og Celarent kan videre ligge til grunn for konsistens- og integritetsjekking i en informasjonsmodell (Sowa 2000). Dersom en for eksempel feilaktig har definert et objekt som en instans av to adskilte klasser, kan dette fanges opp av Ferio-baserte kontroller. Hvis A er en instans av B og C, samtidig som 'Ingen B er C', så er modellen eksempelvis inkonsistent med

hensyn til klasse- versus instansnivå.

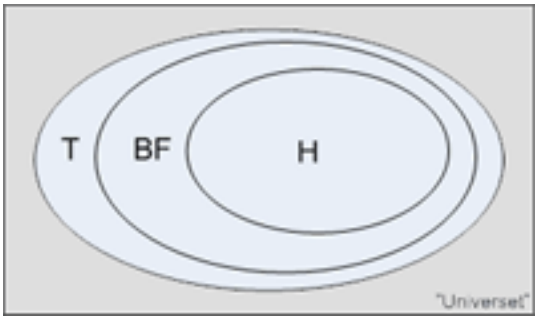
Som nevnt i avsnitt 3.2.2 på side 36 vil en under ontologiutvikling gjerne identifisere klasser som ordnes i et hierarki hvor enkelte klasser faller naturlig inn under andre klasser - for eksempel er 'Stein' modellert som en underklasse av 'Materiale' i Figur 3.2. Klassen 'Stein' angir således mengden M = {x: x er en stein}. Når hvert element i en mengde A også er et element i B, er A en submengde av B (Haggarty 2002). Mengde-submengde relasjonen kalles derfor gjerne for en 'er-en'-relasjon (eng. "is-a"). I Figur 3.2 er klassen 'Stein' modellert som en submengde av klassen 'Materiale' (Stein → er-en → Materiale). Således angir taksonomier 'er-en'-relasjoner, *subsumering* og *instansiering*, i ontologier[9]. Taksonomien i Figur 3.2 kommer klart frem gjennom å følge linjene (er-en) fra subklassene oppad til 'Ting'.

Forholdet mellom mengder i ontologier og taksonomier kan uttrykkes formelt gjennom mengdelærens ekstensjons- og spesifikasjonsaksiom. Ekstensjonsaksiomet fastslår at to mengder er like hvis og bare hvis de inneholder de samme elementene, mens spesifikasjonsaksiomet sier at det for hver mengde A og hver egenskap S finnes en mengde B som består av elementene x fra A som S(x) holder for (B = {x ∈ A: S(x)}) (Halmos 1998). For taksonomien i Figur 3.2 har vi at visse egenskaper avgjør hva som defineres som for eksempel 'Materiale', mens andre egenskaper avgjør hva som defineres som 'Redskap'. Disse to submengdene av 'Ting' består altså av de mengdene av elementer fra 'Ting' som henholdsvis egenskapene ved 'Materiale' og 'Redskap' gjelder for. Fra ekstensjonsaksiomet har vi videre at de to mengdene 'Materiale' og 'Redskap' ikke er like. Taksonomier består således formelt sett av en mengde av mengder utledet fra ekstensjons- og spesifikasjonsaksiomet. En følge av dette er at dersom det opprettes en ny klasse C som klassifiseres som en submengde av en annen mengde B, i en allerede definert taksonomi A, så vil en automatisk få at alle egenskaper ved B (og Bs supermengder) også gjelder for C. Taksonomier er med andre ord autoklassifiserende[10] strukturer. Dersom det i Figur 3.2 legges til en ny klasse 'Pren' under 'Steinredskaper', vil alle egenskaper ved 'Steinredskaper' og 'Steinredskaper's superklasser (oppad til 'Ting'), også gjelde for 'Pren'. Taksonomier kan derfor enkelt utvides, noe som blant annet er avgjørende for for eksempel fletting av emnekart og emnekart-taksonomier (for eksempel indekser), samt

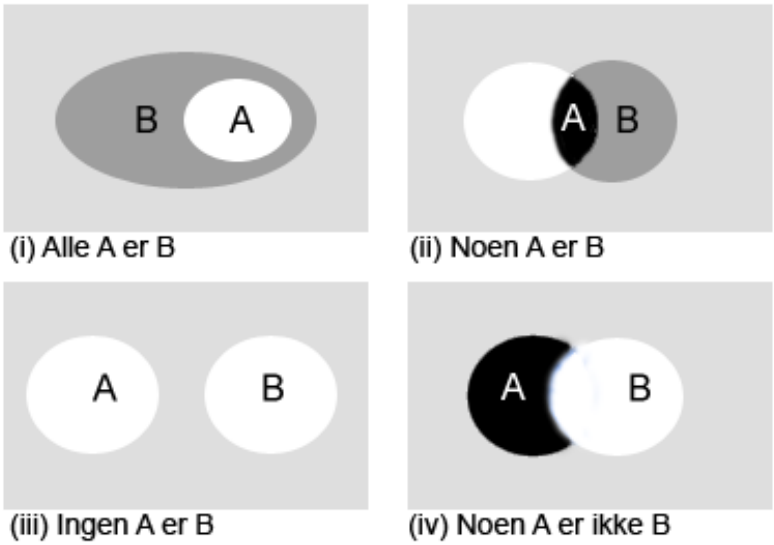
[9]Merk at taksonomier kan ha en flat struktur - for eksempel dersom Figur 3.2 hadde bestått av kun mengdene 'Funn' og 'Heller', og ikke nødvendigvis må inneholde et hierarki av mengder og submengder.

[10]Begrepet 'autoklassifiserende' er her hentet fra Sowas (2000) diskusjon om Minskys frames (se side 22) som blant annet ordner informasjon i klasse-hierarkier.

funksjonen arv i objektorienterte programmeringsspråk.



Figur 3.4: Utdrag av taksonomien fra Figur 3.2, fremstilt i form av et Venn-diagram.



Figur 3.5: Aristoteles' fire kategoriske påstander gjengitt i et Venn-diagram.

Det er vanlig å illustrere mengder og submengder grafisk i Venn-diagram. Figur 3.4 viser et eksempel på et Venn-diagram over deler av taksonomien / ontologien i Figur 3.2. 'Heller' (H) er her modellert som en submengde av 'Bergformasjon' (BF), som igjen er en submengde av 'Ting' (T). Figur 3.5 viser Aristoteles' fire grunntyper av kategoriske påstander i form av et Venn-diagram hvori hver farget sirkel/ellipse angir en mengde x som kan skilles fra andre mengder basert på en mengdespesifikk egenskap S.

Tesaurer

Taksonomier uttrykker som vi har sett subsumering og instansiering. Er-en relasjoner er dog ikke de eneste typene relasjoner som kan eksistere mellom konsepter i en informasjonsmengde; et konkret funn fra en heller vil for eksempel være *funnet ved* en bestemt heller. Tesaurer inneholder informasjon om ulike typer relasjoner mellom elementer i mengder (Jing and Croft 1994). I en tekst-tesaur kan det for eksempel være snakk om relasjoner av typen 'synonym' (ulike ord med lik betydning), 'antonym' (ulike ord med motsatt betydning), eller 'hyponym' (ord hvis mening er inneholdt av andre ord ('subtyper')). Ved hjelp av tesaurer kan man altså identifisere forskjellige elementer med samme, eller ulik betydning. Ikke minst kan man avgjøre på hvilken måte elementer er relaterte til hverandre (Pepper 2000a). På grunn av dette brukes gjerne tesaurer innenfor Information Retrieval, hvor det ofte kan være ønskelig å finne alle relevante ressurser for et bestemt søk, også de dokumenter som inneholder synonymer - en spesiell type relasjon - til ord i en søkefrase.

WordNet[11] er et eksempel på en ontologi som inneholder over 166.000 ord fra det engelske språket, samt relasjoner som 'synonym' og 'antonym' (Miller 1995). Gjennom et grafisk brukergrensesnitt kan en søke i WordNet-databasen, og et søk etter ordet 'run' i WordNet 2.1 vil eksempelvis returnere 41 treff / betydninger. Med utgangspunkt i disse treffene kan en så slå opp for eksempel antonymer, hyponymer, og så videre, til ordet 'run'. Dette er mulig fordi WordNet-databasen inneholder en tesaur over ulike typer relasjoner mellom klasser og instanser.

Informasjon nedfelt i ontologier kan altså ordnes i strukturer som tesaurer for å uttrykke påstander om ulike typer relasjoner mellom elementer.

3.3 Merkespråk for Semantic Web

Mens ontologier gir oss et verktøy for formell beskrivelse av diskursdomener, og taksonomier og tesaurer lar oss ordne, eller "katalogisere", informasjonen i ontologier, behøver vi ytterligere en ingrediens for å kunne utveksle og gjenfinne disse formene for informasjon på tvers av nettverk og applikasjoner. Som nevnt i avsnitt 3.1.1 trengs standardiserte merkespråk for at autonome agenter skal kunne fastslå betydningen til, og handle på grunnlag av, informasjon nedfelt i datasett. For at vi skal kunne bevege oss mot en

[11] WordNet kan lastes ned fra `http://wordnet.princeton.edu/`.

semantisk verdensvev trenger vi derfor standardiserte merkespråk som kan bære informasjon om ontologier, kategorier og -relasjoner.

Språkene beskrevet i dette avsnittet utgjør noen av de best kjente kandidatene for en betydningsorientert verdensvev, og oppfattes gjerne som alternativer til emnekart og XTM. Ettersom applikasjonen utviklet som et grunnlag for denne rapporten er basert på XTM gis det her bare en overfladisk beskrivelse av disse relaterte merkespråkene, mens avsnitt 4.1 gir en mer utfyllende beskrivelse av emnekartteknologien.

3.3.1 Resource Description Framework (RDF)

Resource Description Framework (RDF) (W3C 1998) er et metadataspråk utviklet av W3C og en del av W3Cs Semantic Web initiativ (W3C 2005). RDF er ikke ment å erstatte språk som (X)HTML eller CSS, men er en standard for å uttrykke påstander om ressurser på WWW (Powers 2003). Ved hjelp av RDF kan en altså tilføre WWW informasjon om allerede eksisterende informasjonsressurser, og RDF er derfor en 'information overlay'-teknologi.

RDF er bygget opp av ressurser (eng. "resource"), egenskaper (eng. "properties") og uttrykk (eng. "statements"). En ressurs er et eller annet som vi ønsker å omtale, og identifiseres gjennom en Uniform Resource Identifier (URI). En URI er en tekststreng som unikt identifiserer en abstrakt eller fysisk ressurs (Berners-Lee et al. 1998) - for eksempel en adresserbar "locator" av typen URL, eller ikke-adresserbare identifikatorer som for eksempel ISBN-nummer. Siden URIer er tekststrenger som er garantert å være unike, oppfyller RDF således kravet om utvetydige identitetskriterier for semantiske netteknologier (ref. avsnitt 3.1.1).

Egenskaper angir relasjoner mellom ressurser, og også egenskaper identifiseres ved hjelp av URIer. Et RDF-uttrykk består av en trippel bygget opp av et subjekt, et predikat og et objekt, og angir en egenskap ved en ressurs. Subjektet i en RDF-trippel er en ressurs, predikatet angir relasjonstype, mens objektet utgjør det en ønsker å predikere om subjektet (Powers 2003). Gjennom objektet, hvis type er angitt i predikatet, uttrykker en altså metainformasjon om subjektet. En RDF-trippel (x, R, y) kan derfor tenkes på som en relasjon R (predikat) mellom to objekter x (subjekt) og y (objekt); $R(x, y)$ (Antoniou and van Harmelen 2004).

Påstanden "Hellerprosjektets nettside er basert på XML Topic Maps" kan eksempelvis uttrykkes i form av en RDF-trippel (i):

{http://huin.uib.no/hellerprosjektet,
http://www.example.com/rel.rdf#basertPa, "XML Topic Maps"}

hvor subjektet "Hellerprosjektets nettside" og predikatet "basert på" er representert ved hjelp av (tenkte) URIer. I og med at RDF er "format-uavhengig" er påstanden ovenfor en gyldig RDF-trippel, men for utstrakt utveksling av informasjon over WWW trengs som sagt standardiserte formater og vokabularer. For RDFs vedkommende finnes det blant annet en XML-serialisering; RDF/XML, som tillater dette. En mulig RDF/XML-versjon av (i) er vist i Eksempel 3.3 (se Powers (2003) for beskrivelse av elementer).

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/.../rdf-syntax-ns"
    xmlns:eks="http://www.example.com/ns">

    <rdf:Description
        rdf:about="http://huin.uib.no/hellerprosjektet/index.spy">
        <eks:basertPa>XML Topic Maps</eks:basertPa>
    </rdf:Description>

</rdf:RDF>
```

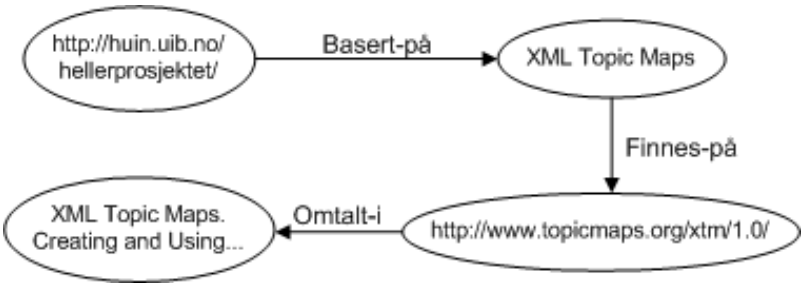
Eksempel 3.3: RDF-trippel som angir at "Hellerprosjektets nettside (angitt ved URI) er basert på XML Topic Maps". URIer og XML namespace'et (xmlns) eks er 'konstruert' for eksempelets skyld.

Det essensielle ved RDF er at en beskriver egenskaper ved ressurser i form av tripler, og at en gjennom bruk av URIer utvetydiggjør[12] omtalte ressurser. Ettersom et RDF-dokument består av en mengde tripler, og fordi en kan referere til ressurser ved hjelp av deres URIer, er dessuten den innbyrdes rekkefølgen til triplene i et RDF-dokument vilkårlig. Dette gjør at en i RDF/XML ikke er like strengt bundet til XMLs trestruktur,

[12]For RDFs vedkommende er ikke dette 100% korrekt. Mer om dette i avsnitt 4.1.1, s. 55.

som i vanlig XML hvor relasjoner uttrykkes gjennom en påtvunget nøsting av elementer (RDF/XML kan dog selvsagt ikke bryte med regler for XML). Et RDF-fragment må ikke nødvendigvis inneholdes av andre RDF-fragmenter, men kan referere til disse ved hjelp av URIer. Oppbygningen av RDF i tripler medfører dessuten at informasjonsstrukturer merket i RDF av natur er assosiative. Med utgangspunkt i en 'node' i et RDF-dokument kan en bevege seg til alle relaterte noder og ressurser, de relaterte nodenes relaterte noder og ressurser, og så videre. Dette lar seg lett illustrere i såkalte RDF-grafer, eller i direkte grafer (di-graf).

Øvre del av Figur 3.6 viser en di-graf av Eksempel 3.3. Kantene går her fra ressursen (subjektet) til verdien (objektet). Dersom RDF-biten fra Eksempel 3.3 var utvidet, for eksempel ved å inkludere RDF-uttrykk om XML Topic Maps, ville en med andre ord kunnet bevege seg i et assosiativt nettverk av ressurser (innenfor KI kalt et "Semantisk nett" (Antoniou and van Harmelen 2004)) og følge forbindelser med utgangspunkt i informasjonsstrukturens oppbygning. Slik kan en for eksempel ved hjelp av RDF konstruere informasjonsmodeller og grensesnitt som kan benyttes for å la brukere utforske informasjonsressurser basert på den logiske oppbygningen av en ontologi, noe som kanskje kan sies å være i tråd med Bushs tanker om assosiativ linking i Memex (ref. avsnitt 2.1.1).



Figur 3.6: RDF-tripler som semantisk nett. Ved å følge pilene kan en bevege seg mellom assosierte informasjonsressurser.

Oppgaven vil ikke gå i ytterligere detalj omkring oppbygningen av RDF og RDF/XML, men henviser leseren til Powers (2003) for utfyllende informasjon om både RDF/XML og RDF generelt.

RDF Schema

Ved hjelp av RDF kan en altså uttrykke påstander om hva som helst - også andre RDF-uttrykk, men RDF i seg selv tillater en ikke å spesifisere typer og relasjoner, eller ontolo-

gier. For å kunne beskrive egne domener i RDF, og for at RDF etter vår definisjon skal kunne kalles en semantisk netteknologi (ref. avsnitt 3.1.1), behøves derfor en utvidelse. RDF Schema (RDFS) (W3C 2004c) lar en definere typer og egenskaper ved typer, og fungerer således som et begrenset ontologispråk for RDF. RDFS er altså ikke et skjema på samme måte som XML Schema, som lar en angi datatyper og begrensninger på XML-vokabularer, men muliggjør opprettelse av domene-spesifikke RDF-vokabularer (Powers 2003). Alle domene-spesifikke RDF-elementer bygger på en begrenset mengde "kjerne-elementer" spesifisert i RDFS 'standarden', som for eksempel rdfs:Resource og rdfs:Class. RDFS angir også elementer som rdfs:subClassOf, rdfs:subPropertyOf, rdfs:range og rdfs:domain (Powers 2003). Med utgangspunkt i RDFS kan en altså beskrive klasser og taksonomier (klasse-subklasse relasjoner) for et spesifikt RDF-vokabular.

3.3.2 Web Ontology Language (OWL)

Til tross for at en ved hjelp av RDFS kan opprette klasser og taksonomier, kan ikke RDF/RDFS (RDF(S)) bære informasjon om alle sider ved ontologier. I RDF(S) kan en blant annet ikke angi gyldighetsområder for egenskaper, hvorvidt klasser er adskilte eller ikke, eller kombinere klasser ved hjelp av operatorer som union, snitt og komplement. En kan heller ikke legge begrensninger på antallet tillatte verdier for bestemte egenskaper (kardinalitetsbegrensning), eller spesifisere spesielle karakteristikker ved egenskaper - for eksempel hvorvidt en egenskap er transitiv eller invers av en annen - i RDF(S) (Antoniou and van Harmelen 2004). Med utgangspunkt i problemer som disse har W3C utviklet et komplekst ontologispråk; Web Ontology Language (OWL) (W3C 2004b), som gjør det mulig å uttrykke nettopp slike aspekter ved ontologier.

I tillegg til å tillate opprettelse av klasser og taksnomier, har en i OWL standardiserte måter for å angi at klasser er adskilte (owl:disjointWith), operasjoner som snitt (owl:intersectionOf), egenskaper som transitivitet (owl:TransitiveProperty), og så videre. At denne typen informasjon er nedfelt i selve standarden gjør at en eksplisitt og entydig kan angi ytterligere aspekter ved ontologier, noe som gjør en bedre sikret mot at deler av ens informasjonsmodeller tolkes forskjellig i ulike applikasjoner. OWL er dessuten bygget opp som en kombinasjon av RDFS og Description Logic (DL) (Antoniou and van Harmelen 2004), hvilket gjør det til et svært uttrykkskraftig språk. Grunnet kompleksiteten dette medfører, er OWL delt inn i 3 "underspråk": OWL Full er maksimalt uttrykkskraftig og inkluderer alle språkkonstrukter; OWL DL legger begrensninger på

hvordan språkkonstruktene fra OWL Full kan benyttes, men er komplett med hensyn til applikasjoners resonnersingsevne (alle konklusjoner kan beregnes, noe som ikke er garantert i OWL Full); OWL Lite legger ytterliggere begrensninger på OWL DL og gjør språket mindre uttrykkskraftig, men lettere for menneskelige brukere å jobbe med (W3C 2004b).

Vi har nå sett hva en semantisk vev er, hva som kjennetegner semantiske netteknologier og hvordan en ved hjelp av semantiske netteknologier som OWL kan flytte større deler av semantikken fra applikasjon- til dokumentnivå. Neste kapittel vil ta for seg emnekart og en konkret implementasjon av en semantisk netteknologi.

Kapittel 4

Implementasjon

Teorien diskutert i oppgavens del 1 (kapittel 1-3) har som innledningsvis nevnt dannet utgangspunkt for det praktiske arbeidet utført som en del av dette mastergradsprosjektet. Denne siste delen av rapporten søker å vise hvordan kunnskapen fra del 1 er anvendt i utviklingen av publiseringsløsningen for Hellerprosjektet. Nettstedet finnes på URL `http://huin.uib.no/hellerprosjektet/`. All kode finnes på vedlagte CD-ROM (se appendiks A for utdrag av XTM).

Ettersom det for Hellerprosjektet er benyttet en annen løsning enn RDF(S)/OWL, gjøres det i dette kapittelet først kort rede for valgte teknologi - emnekart, med vekt på emner, assosiasjoner, forekomster og identitet. Deretter diskuteres noen forskjeller mellom XML Topic Maps og RDF(S)/OWL, før oppgaven beveger seg inn på den konkrete implementasjonen av emnekart i Hellerprosjektets vevapplikasjon. Det gjøres her rede for tanker bak valg tatt under utviklingen av applikasjonen, samt muligheter for videreutvikling, bruksområder og forbedringer. Da det er i egenskap av å være en - etter vår definisjon (se avsnitt 3.1.1) - semantisk netteknologi at emnekartstandarden i denne sammenheng er interessant, etterstrebes det her ikke nødvendigvis et strengt skille mellom standardens abstrakte modell og dens XML-serialisering. Kapittelet søker ikke å være en teknisk innføring i emnekart eller XML Topic Maps, og selv om eksempler gis i form av XTM, henvises det til ISO 13250 (ISO 2002) og XTM-spesifikasjonen (Pepper and Moore 2003) for detaljer omkring begreper som ikke tas opp i teksten (dyptgående om semantikken bak elementer, perspektivering (scope), varianter av emnenavn (variant), etc.).

4.1 Emnekart

Emnekart-idéen har utspring i Davenport-gruppen (UNIX-produsenter med flere) sitt behov for å produsere samlende hovedindekser, glossarer og innholdsfortegnelser av store mengder uavhengig dokumentasjon (Pepper 2000a). Denne gruppen ønsket å flette store og uavhengige dokumentmengder, samtidig som inkonsistens og overflødighet i resulterende informasjonsmengde måtte holdes på et absolutt minimum (Park and Hunting 2003). De ulike dokumentasjonene ble dessuten oppdatert svært hyppig, noe som også måtte gjenspeiles i for eksempel indekser og fletting av indekser. Disse organisasjonene hadde altså behov for en teknologi som tillot representering av egenskaper ved de ulike dokumenter og indekser - for eksempel eksplisitt merking av spesielle ord, forekomster av ord og ulike typer relasjoner mellom ord (som i tesaurer, ref. avsnitt 3.2.3). En slik teknologi måtte derfor kunne skille mellom ulike typer informasjon og støtte gjentatt fletting av informasjonsressurser hvor like konsepter slås sammen og urelaterte konsepter forblir urelaterte. Teknologien måtte med andre ord være basert på metainformasjon og utvetydige identitetskriterier, samt støtte metoder for opprettelse av 'live'-indekser.

Arbeidet med utvikling av en slik teknologi ble opprinnelig ledet av Davenport-gruppen, som i prosessen blant annet utviklet DocBook[1] (Pepper 2000a) for merking av dokumenttyper som bøker og artikler. Arbeidet ble senere videreført av IDEAlliance under arbeidstittelen "Conventions for the Application of HyTime"[2] (Pepper and Moore 2003), og munnet i 1999 ut i en formalisert internasjonal standard, ISO/IEC 13250, for organisering og representering av informasjon og informasjonsressurser.

4.1.1 Bestanddeler

I ISO/IEC 13250, Topic Maps (Second Edition) (2002) presenteres emnekartstandarden som en standard notasjon for å representere informasjon om strukturen til informasjonsressurser som beskriver emner og relasjoner mellom emner (kalt assosiasjoner), mens et emnekart er en samling dokumenter som benytter seg av notasjonen slik den er definert i ISO/IEC 13250 (ISO 2002, s. iii).

I emnekart er informasjon strukturert rundt emner, assosiasjoner og forekomster.

[1] DocBook var opprinnelig et SGML-vokabular / en SGML Document Type Definition, men finnes i dag også i form av blant annet XML Schema. Se Walsh (2005) for mer informasjon om DocBook.

[2] HyTime er en kompleks SGML-basert hypertekststandard publisert av ISO/IEC.

Disse konseptene er sterkt relatert til emnekart-idéens opprinnelse, ettersom det var gjennom arbeidet som førte frem til emnekartstandarden at en identifiserte ulike (generelle) elementer som går igjen i tekster og indekser. Indekser kan for eksempel ses på som en samling av nettopp disse tre hovedtypene konsepter, hvor emner tilsvarer klasser og instanser av ord, assosiasjoner tilsvarer referansetyper som 'se også', og forekomster utgjør de ulike sidehenvisninger (Pepper 2000a). Videre bygger standarden på utvetydige identitetskriterier som muliggjør 'nøyaktig' sammenfletting og adskillelse av emner og relasjoner, og dermed også indekser. Bakgrunnen som indekseringsteknologi gjør dessuten emnekart svært egnet for å modellere informasjon på måter som gjør den lett navigerbar, akkurat som bak-i-boken-indekser kan sies å representere kart over innholdet i bøker (Pepper 2000a). Emnekartstandarden har med andre ord sitt utspring i dokumenthåndtering og -indeksering, men egner seg like fullt for merking av arkeologisk forskningsmateriale som for merking av tekster for indekseringsformål. Dette da standarden for eksempel ikke legger føringer på hvilke typer klasser, objekter og relasjoner som kan uttrykkes. Det gis i det følgende en oversikt over hovedbestanddelene i emnekart.

Emner

Emner representerer objekter (i emnekartterminologi kalt subjekter), i vid forstand, og tilsvarer derfor elementer i en ontologi. I emnekartstandarden merkes emner som emner (XTMs topic-element), og et emne kan representere "any 'thing' whatsoever [...] about which anything whatsoever may be asserted by any means whatsoever." (Pepper 2000a, s.10). Å representere abstrakte begreper fra Platons idéverden i form av ulike emner i et emnekart er eksempelvis ikke nødvendigvis mer utfordrende enn å representere det konkrete subjektet (personen) 'Platon'.

En viktig side ved emner er at de kan defineres som klasser, eller som instanser av en eller flere klasser (også angitt ved emner), noe som tillater en å på en eksplisitt og standardisert måte angi ulike typer, og instanser (XTMs instanceOf-element), av konsepter i et diskursdomene. I tillegg til dette kan en blant annet tildele emner ulike navn, hvor de ulike navn også er merket som navn. Det vil si at det i emnekartstandarden er bygget inn semantikk relatert til navn, og at en derfor kan uttrykke lingvistiske forhold som synomymi og homonymi. Videre kan emner ha ulike forekomster og emneidentitet. Også forekomster og identitet merkes med egne typer merker, hvilket gjør det mulig for både datamaskiner og menneskelige brukere å enkelt avgjøre hva som er en forekomst og hva som utgjør et gitt emnes identitet.

Eksempel 4.1 viser hvordan en i emnekartstandardens XML-syntaks, XML Topic Maps, kan definere et emne med navnene "Hellerprosjektet" og "The Heller Project". De to navnene er å regne som synonymer, ettersom begge refererer til ett og samme subjekt (Garshol 2004). Ved hjelp av et slikt eksplisitt skille mellom relaterte navn kan en gjøre informasjon lettere tilgjengelig, idet et søk i en dokumentmengde hvor en kan identifisere synonymer lettere kan produsere relevante resultater (ref. diskusjon om fulltekstsøk, s. 25).

```
<topic id="hellerprosjektet">
  <baseName>
    <baseNameString>Hellerprosjektet</baseNameString>
  </baseName>
  <baseName>
    <scope> <topicRef xlink:href="#english"> </scope>
    <baseNameString>The Heller Project</baseNameString>
  </baseName>
</topic>
```

Eksempel 4.1. XTM av et emne med to ulike navn. "Hellerprosjektet" vil alltid være et gyldig navn for dette emnet (navnet befinner seg i "unconstrained scope"), mens et alternativt navn - "The Heller Project" - kun er gyldig under perspektivet "english" (eller rettere sagt: under perspektivet angitt i emnet med ID "english").

Assosiasjoner

Assosiasjoner angir relasjoner mellom emner i emnekart (ISO 2002). Assosiasjoner uttrykkes gjennom egne sett av assosiasjons-merker, hvor en blant annet eksplisitt angir hvilke emner som deltar i en gitt assosiasjon. I emnekartstandarden legges det ikke føringer på hvilke typer assosiasjoner som kan defineres, noe som gir rom for å uttrykke assosiasjoner av en hvilken som helst aritet.

Som for emner kan også assosiasjoner types, det vil si angis som en instans (XTMs instanceOf) av egendefinerte emner (assosiasjonstyper). Man kan for eksempel definere assosiasjoner av typen *funnetved* (relasjonseksempelet på side 35), eller hvilke som helst andre relasjoner. Dette gjøres ved å la et emne representere klassen *funnetved* og deretter

spesifisere assosiasjoner som instanser av denne klassen.

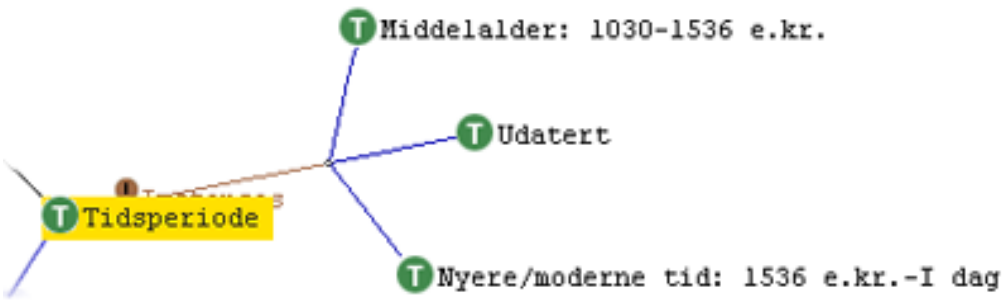
Emnekart mangler en standardisert løsning for å eksplisitt uttrykke domene og rekkevidde i relasjoner av aritet > 2, men lar en perspektivere relasjoner gjennom såkalte assosiasjonsroller. Assosiasjonsroller lar en eksplisitt uttrykke rollen til et emne som deltar i en relasjon, noe som gjør at en for eksempel kan si at et funn er $funnetved$ en heller og at helleren er $funnplassfor$ funnet (Eksempel 4.2). Dette gir oss altså en metode for å unngå å si at begge emnene er $funnetved$ hverandre.

```
<association id="funnet_ved_assoc_01">
  <instanceOf>
    <topicRef xlink:href="#funnet_ved_assoc" />
  </instanceOf>
  <member>
    <roleSpec> <topicRef xlink:href="#funn" /> </roleSpec>
    <topicRef xlink:href="#pil_ab21" />
  </member>
  <member>
    <roleSpec> <topicRef xlink:href="#funnplass" /> </roleSpec>
    <topicRef xlink:href="#savarhelleren" />
  </member>
</association>
```

Eksempel 4.2. XTM-assosiasjon som angir en konkret "funnet_ved"-assosiasjon. member-elementene angir deltakeremner, mens roleSpec uttrykker spesifikke assosiasjonsroller. Emner som er referert til vha. topicRef bør eksistere annetsteds i dokumentet.

Assosiasjoner lar en altså relativt enkelt beskrive for eksempel taksonomier og tesauruer. Lik RDF er også emnekart assosiative informasjonsstrukturer, idet de enkelte delene er relaterte til hverandre og gjør det mulig å bevege seg mellom relaterte informasjonsfragmenter. En mulig grafisk fremstilling av emner og forholdet mellom dem - i dette tilfellet instansiering / er-en relasjoner - er fremstilt i Figur 4.1, som viser deler av Hellerprosjektets XTM visualisert gjennom programmet TMNav[3].

[3]Tilgjengelig på `http://www.tm4j.org/tmnav.html`.



Figur 4.1: Grafisk fremstilling av assosiativitet i emnekart vha. TMNav. Del av ontologi under utvikling av Jan Erik Mandelid.

Forekomster

I emnekartstandarden defineres forekomster, eller "topic occurrences", som "information that is relevant to a given subject" (ISO 2002, s. 6). Forekomster angir med andre ord - uavhengig av informasjonstype og adresseringsmekanisme - en spesialisert og standardisert notasjon for relasjoner mellom et subjekt og informasjon om subjektet (ISO 2002). I likhet med emner og assosiasjoner kan også forekomster types gjennom å merkes som instanser av klasser (emner).

Forekomster kan være lagret direkte i emnekartet - for eksempel en tekstlig beskrivelse, eller en kan referere til eksterne ressurser. Skillet mellom interne og eksterne forekomster er nedfelt i standarden gjennom et eget sett av forekomst-merker.

```
<topic id="heller">
  <!– emnenavn og lignende her –>
  <occurrence>
    <instanceOf> <topicRef xlink:href="#bilde" /> </instanceOf>
    <resourceRef xlink:href="http://example.com/heller.jpg" />
  </occurrence>
</topic>
```

Eksempel 4.3. Forekomst av et emne (XTM). I tillegg til resourceRef-elementet, som refererer til ressurser, inneholder XTM-standarden et element, resourceData, som lar en lagre informasjonen om emner direkte i XTM-dokumenter.

Subjektidentitet

I et emnekart spiller identitieten til subjektene som emnene representerer en sentral rolle. Å gjøre en maskin, og også et menneske, i stand til å fastslå at identiteten til et emne A er subjektet B, er avgjørende for om Semantic Web-visjonen skal kunne lykkes. Dette fordi å kunne etablere betydning forutsetter et utvetydig identitetskriterium (ref. definisjon i avsnitt 3.1.1). Dersom to agenter på en betydningsorientert vev skal kunne utveksle informasjon om noe, må de kunne avgjøre at de utveksler informasjon om det samme 'noe'. Mens ontologier tilbyr oss en metode for å utvetydiggjøre identiteten til objekter innen et diskursdomene, vil en på en betydningsorientert vev også være avhengig av å utvetydig kunne fastslå hvorvidt to 'ting' med samme navn eller samme adresse er like, eller ikke. I emnekart er løsningen på dette problemet subjektidentitet; 'Subject Identity'.

Identiteten til et emne som representerer 'Sævarhelleren' kan for eksempel fastslåes ved at emnets subjektidentitet gjennom egne merker settes til å være identifiserbart via en HTML-side som beskriver Sævarhelleren, og som finnes på en bestemt URI. Dette gjøres også i andre teknologier, for eksempel W3Cs RDF (ref. avsnitt 3.3.1). Et vesentlig aspekt ved emnekarts subjektidentitet er dog at en her har standardiserte merker for å skille mellom hvorvidt et emne representerer selve informasjonsressursen (HTML-dokumentet), eller ett eller annet som omtales i, er indikert ved, informasjonsressursen. Mens RDF ikke lar en skille mellom disse typene identitiet, har ISO og TopicMaps.org i ISO 13250 og XTM 1.0 innført et skille mellom "addressable subjects" og "non-adressable subjects" (Pepper and Schwab 2002, s. 6). Selve HTML-dokumentet som beskriver Sævarhelleren er et eksempel på et adresserbart subjekt - det kan addresseres ved hjelp av en URI. Helleren Sævarhelleren befinner seg dog i Herand i Hardanger, ikke på en server i Bergen. Sævarhelleren kan derfor ikke hentes ned via en URI, og er med andre ord en ikke-adresserbar ressurs. Ettersom det eksisterer et HTML-dokument som beskriver Sævarhelleren kan en dog indikere identiteten til subjektet 'Sævarhelleren' ved å peke til HTML-dokumentet som beskriver denne helleren. Som sagt lar dette seg gjøre i RDF, men mens en i RDF ikke eksplisitt angir hvilken type identitet det er snakk om - og en RDF-applikasjon derfor ikke kan avgjøre om to referanser faktisk peker til det samme subjektet - skiller altså emnekart mellom ulike typer identitet, med tilhørende merker. Eksempel 4.4 viser hvordan to emner i XTM kan gis ulik identitet - til tross for at identiteten er angitt ved samme URI. Det øverste emnet representerer subjektet Sævarhelleren, og emnets identitet indikeres gjennom bruk av elementet subjectIndikatorRef. En menneskelig leser kan på `http://huin.uib.no/hellerprosjektet/news.spy?newsid=10` lese seg

til hva emnet representerer, mens en datamaskin på sin side kan fastslå at emnet representerer noe som er tilgjengelig via URIen, og ikke ressursen i seg selv. Emnet med ID savarhtml representerer derimot selve HTML-dokumentet, noe som kommer klart frem gjennom bruk av resourceRef-elementet som benyttes for å referere direkte til ressurser.

```
<!– Sævarhelleren –>
<topic id="savarhelleren">
  <subjectIdentity>
    <subjectIndicatorRef
    xlink:href="http://huin.uib.no/hellerprosjektet/news.spy?newsid=10" />
  </subjectIdentity>
  <!– navn og forekomster ->
</topic>

<!– HTML-dokumentet –>
<topic id="savarhtml">
  <subjectIdentity>
    <resourceRef
    xlink:href="http://huin.uib.no/hellerprosjektet/news.spy?newsid=10" / >
  </subjectIdentity>
  <!– navn og forekomster –>
</topic>
```

Eksempel 4.4. URI som subjektindikator versus URI som subjektidentitet.

Publiserte subjekter

Enhver informasjonsressurs (både off- og on-line) kan i praksis benyttes som en subjektindikator og angi subjektidentitet for ulike emner. Et publisert subjekt er derimot et subjekt som kan indikeres gjennom en URI (med tilhørende subjekt-beskrivelse) som er gjort offentlig tilgjengelig med det formål å muliggjøre utvetydige, stabile, pålitelige og autoritative subjekter, subjektindikatorer og subjektidentifikatorer (Pepper 2003).

Formålet med, og behovet for, publiserte subjektindikatorer (PSIer) er å støtte utveksling, gjenbruk av og konsistens i, og mellom, ontologier. Ved å tilby og benytte seg av PSIer sikrer en at emner i ulike emnekart kan dele en felles - og entydig - subjekti-

dentitet. Dermed kan både mennesker og dataprogrammer fastslå at ulike emner og emnekart representerer de samme, eventuelt forskjellige, subjekter. For eksempel tilbyr TopicMaps.Org PSIer for blant annet land og språk, mens selve XTM-spesifikasjonen inneholder PSIer for en mengde kjernekonsepter - for eksempel `http://www.topicmaps.org/xtm/1.0/core.xtm#superclass-subclass` som angir "The core concept of superclass-subclass; the class of association that represents superclass-subclass relationships between topics." (Pepper and Moore 2003, p. 2.3.2).

Reifisering

Ettersom et emne kan representere hvilket som helst subjekt kan det også representere et emnekart eller karakteristikker ved andre emner. I emnekartstandarden kalles denne 'tingliggjørings'-prosessen, samt prosessen med å la et emne representere et subjekt, reifisering[4]. Reifisering er således et viktig aspekt ved emnekartstandarden, og ved å la subjektidentiteten til et emne referere til et annet emnes karakteristikker kan en enkelt reifisere andre emnekart-elementer. Ved å gjøre en emnekarakteristikk gjenstand for reifisering kan en for eksempel gi assosiasjoner navn, eller tilføre informasjon til emnenavn eller forekomster. I applikasjonen utviklet for Hellerprosjektet er reifisering blant annet benyttet for å tilføre informasjon (bildetekst) til bildeforekomster (se appendiks A for eksempler).

Fletting

Subjektidentitet spiller også en avgjørende rolle ved fletting (eng. "merging") av emnekart. Når to eller flere emnekart (og emner) flettes, blir ulike emner som omtaler samme subjekt slått sammen til ett nytt emne hvis karakteristikker (navn, forekomster, assosiasjonsroller, og så videre) er lik unionen av de 'deltakende' emnenes karakteristikker (Pepper and Moore 2003). Dersom to eller flere emners identitet er angitt ved en og samme subjektindikator, antas det eksempelvis at de uttrykker påstander om ett og samme subjekt. Fletting reduserer dermed mengden redundans i emnekart - samler informasjonen på ett sted og fjerner overflødigheter, og kan bidra til å sikre konsistens over en mengde emnekart (ISO 2005). Fletting tilbyr også en metode for å "inkludere" ekstern informasjon i et emnekart, for eksempel gjennom å flette et XTM-dokument med

[4]ISO/IEC (2005) unngår eventuelle filosofiske diskusjoner gjennom å si at "[...] the term 'reification' in this part of ISO/IEC 13250 is not to be confused with its use in philosophy." (ISO 2005, s. 9).

andre XTM-dokumenter.

4.2 Publiseringsløsningen

4.2.1 Valg av teknologi

Som nevnt i oppgavens avsnitt 3.3 anses gjerne emnekart og XTM for å være alternativer til teknologiene RDF(S) og OWL. Likheten mellom teknologiene er da også til dels innlysende. For eksempel representerer og uttrykker begge teknologier påstander om konsepter i diskursdomener (Garshol 2004), og benyttes i stor grad for å tilføre metainformasjon til ressurser på WWW. Videre er det mulig å 'mappe' informasjon fra for eksempel RDF/XML til XTM - en prosess beskrevet av Garshol (2003) og Pepper og Garshol (2002)[5] - noe som tilsier at teknologiene i alle fall til en viss grad må være relaterte til hverandre. Det eksisterer dog også klare forskjeller mellom eksempelvis emnekart og RDF. Alle RDF-uttrykk består av en trippel, og RDF støtter bare 2-plass relasjoner (ref. avsnitt 3.3.1). Emnekart støtter derimot n-ære relasjoner, og er bygget opp av emner med en tilhørende mengde standardiserte karakteristikker. Emnekartstandarden skiller dessuten mellom adresserbare subjekter - identifisert ved URIer - og ikke-adresserbare subjekter - identifisert ved URIer som subjektindikatorer, noe RDF(S)/OWL ikke gjør. På den annen side er semantikken for relasjoners domene og rekkevidde, adskilte mengder og andre egenskaper innebygget i RDF(S)/OWL, mens emnekartstandarden - til tross for sine assosiasjonsroller - mangler en standardisert notasjon for slike aspekter ved ontologier. Adskilthet med mer kan riktignok uttrykkes ved hjelp av egendefinerte relasjonestyper og/eller PSIer, men semantikken i emnekartstandarden er relativt svak med hensyn til relasjoner med aritet > 2 - noe som medfører en viss fare for at ulike applikasjoner tolker like relasjoner på forskjellige måter.

Forskjellene ved RDF(S)/OWL og emnekart har sannsynligvis opphav i teknologienes ulike utgangspunkt. Steve Pepper summerer dette opp i argument nr. 3 i teksten *Ten Theses on Topic Maps and RDF*[6]:

> Topic mapping has its roots in traditional finding aids such as back-of-book

[5] Firmaet Ontopia AS, hvor nevnte forfattere har sitt virke, har endog publisert en rapport; "RDF to topic maps mapping" (Ontopia 2003).

[6] For en diskusjon av forskjeller mellom, og integrering av, emnekart og RDF, se Garshol (2003); Lacher and Decker (2001).

> indexes, glossaries and thesauri. RDF has its roots in formal logic and mathematical graph theory. Topic mapping is knowledge representation applied to information management from the perspective of humans. RDF is knowledge representation applied to information management from the perspective of machines (Pepper 2000b).

Dersom en sammenligner (den åpenbare) kompleksiteten i et innholdsrikt OWL-dokument (se Antoniou og van Harmelen (2004) for eksempler) med kompleksiteten av et innholdsrikt XTM-dokument, vil en nok relativt fort kunne si seg enig i denne påstanden. XTMs innebygde semantikk for navn, forekomster og instansiering har utspring i et behov for indeksering og navigering i informasjonsmengder, og gjør at informasjonsfragmenter i større grad samles på færre steder (et emne er således en 'container' for karakteristikker ved et subjekt) enn hva som er tilfellet i OWL. OWL er i større grad utformet for å være et generelt semantisk merkespråk med støtte for formell logikk, med den kompleksiteten det medfører. Kompleksiteten til RDF(S)/OWL gir stor uttrykkskraft og gode forutsetninger for å merke de fleste typer informasjon og ontologier, men at all informasjon stykkes opp i enkelt-elementer som tripler gjør det til en viss grad vanskelig å jobbe (manuelt) med RDF- og OWL-dokumenter. Emnekart er på sin side lettere for mennesker å jobbe med, men det "menneskevennlige"-aspektet ved emnekart og XTM kan muligens gjøre det vanskelig å formelt uttrykke enkelte sider ved et diskursdomene.

Hellerprosjektet og XTM

For Hellerprosjektets vedkommende har styrkene ved å bruke XTM versus OWL blitt ansett som større enn svakhetene, hvilket er grunnen til at XTM er valgt som merkespråk over OWL. Prosjektet har i hovedsak vært opptatt av å kunne klassifisere informasjon i et betydningsorientert merkespråk. Dette for å kunne bevare informasjon knyttet til klasser, instanser og taksonomier, da løsningen utviklet for Hellerprosjektet skal danne grunnlag for utforsking av, og navigering i, forskningsmateriale fra prosjektet. Det var dessuten ønskelig at denne typen informasjon og metainformasjon kunne nedfelles i et åpent format som muliggjøre utveksling og gjenfinning av informasjon på tvers av ulike applikasjoner. Videre ønsket en å bruke en teknologi med støtte for autoklassifikasjon, da en ved hjelp av egenskaper ved slike strukturer enkelt kan utvide sine informasjonsmodeller. I emnekart oppnås dette blant annet gjennom opprettelse av eksplisitt uttrykte taksonomi-assosiasjoner (ref. diskusjon om taksonomier, side 39) og instansiering

(instanceOf). Legges et nytt emne til et emnekart, vil det automatisk finne sin plass i strukturen og arve egenskaper fra sine superklasser.

OWL er svært uttrykkskraftig, men i enkelte tilfeller vil en ikke ha behov for all uttrykkskraft som befinner seg i et så komplekst språk som OWL (noe som for øvrig gjenspeiles i OWLs tre-deling (Antoniou and van Harmelen 2004)) - så også for Hellerprosjektet. Samtidig gjelder det at prosjektet har lagt vekt på at det skal være mulig å legge til rette for brukertilpasset presentasjon av informasjon, noe som er lett tilgjengelig via perspektivering i XTM. At XTM er / synes lettere enn OWL å jobbe med kan også sies å være en fordel for prosjektet, spesielt med tanke på en eventuell innlæringskurve for 'arvtakere' av prosjektets publiseringsløsning.

4.2.2 Emnekart i PyToM

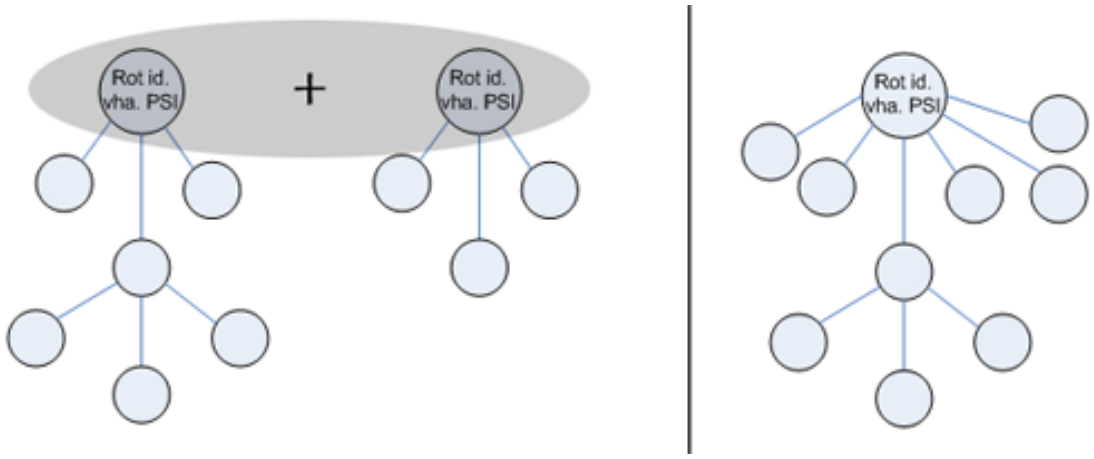
Vi har nå sett hva et emnekart er, og hvorfor Hellerprosjektet valgte å basere sin publiseringsløsning på XML Topic Maps. Dette avsnittet presenterer oppbygning av emnekartmotoren benyttet i vevapplikasjonen, samt ulike valg tatt underveis i utviklingen av prosjektets XTM-drevne nettsted.

Emnekartmotoren, eller parseren (tolkeren), som benyttes som en del av Hellerprosjektets publiseringsløsning er utviklet av Audun Stolpe ved Seksjon for humanistisk informatikk. Programmet, foreløpig kalt PyToM, er skrevet i programmeringsspråket Python[7]. For parsing av XTM baserer PyToM seg på offentlig tilgjengelige kode-bibliotek. En spesiell PyToM-klasse tar seg av tolking og bygging av emnekartrepresentasjoner, og når PyToM parser et emnekart, vil alle emnekartkonsepter representeres av objekter fra tilhørende Python-klasser. Et konkret emnekart representeres for eksempel av et emnekartobjekt (PyToM.TopicMap), et emne av et emneobjekt (PyToM.TypedElements.Topic), en assosiasjon av et assosiasjonsobjekt (PyToM.TypedElements.Association), og så videre. Disse aspektene ved PyToM kan sies å være grunnleggende, og vil derfor ikke diskuteres videre. Det for oss interessante er hvordan emnekart og relasjoner, og dermed ontologier, representeres internt i motoren.

I PyToM er alle klasser representert som en submengde av en felles superklasse identifisert ved en PSI. Alle emner som ikke er merket som instanser av andre emner linkes direkte til denne predefinerte taksonomiske roten. Samtlige emner i et emnekart vil med andre ord i PyToM representeres som underemner av en felles rot. Utad vil rotemnet dog

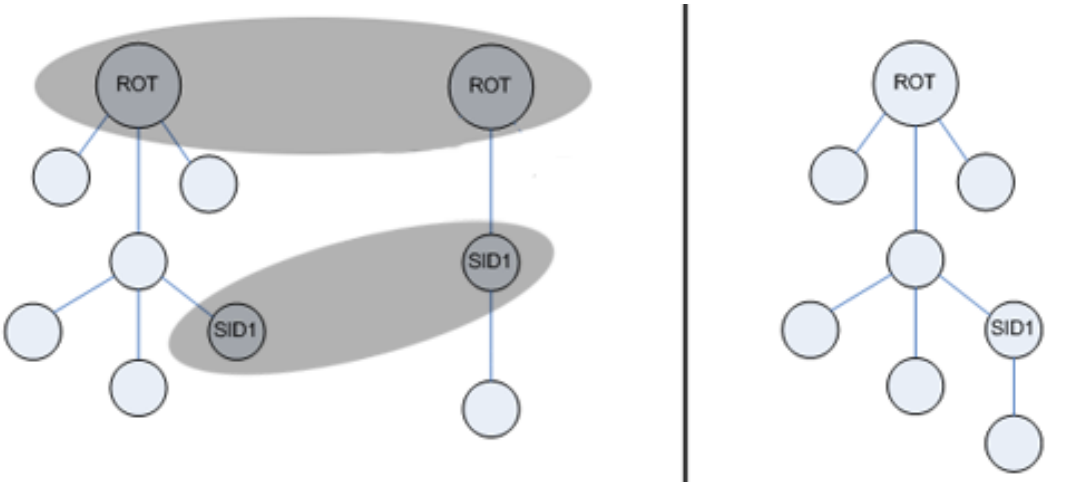
[7] `http://www.python.org`

ikke være synlig, og dets eneste rolle er således å være et felles 'knutepunkt' for emner internt i motoren. Dette betyr blant annet at selv om en ikke spesifiserer en taksonomi i XTM-dokumenter, så vil alle emner være klassifisert i en (så å si flat) automatisk generert taksonomi med nevnte superklasse som rot (internt i PyToM). Definerer emnekartet en taksonomi, vil også denne falle inn under den interne taksonomiske roten. Programmet kan derfor lett bevege seg mellom emner og ha kontroll på hvor de enkelte emnene befinner seg. Samtidig gir dette oss en enkel løsning med hensyn til fletting av emnekart, ettersom roten fra ulike emnekart alltid vil være identifisert gjennom en felles PSI og derfor kan slåes sammen til ett emne. Ved en slik sammenslåing vil den nye roten inneha rollen som superklasse for samtlige emner fra samtlige emnekart. Figurene 4.2 og 4.3 illustrerer fletting av emnekart i PyToM. Figurenes venstre illustrerer den taksonomiske ordningen før fletting, mens høyre del illustrerer den nye representasjonen av de to emnekartene i en felles PSI-identifisert taksonomisk rot. I Figur 4.2 flettes kun de taksonomiske røttene i PyToM, mens emnekartene i Figur 4.3 også inneholder emner som flettes på grunn av identisk subjektidentitet.



Figur 4.2: Fletting av emnekart i PyToM mht. felles PSI-identifisert taksonomisk rot.

Ettersom PyToM er 'under utvikling' har det vært nødvendig å gjøre enkelte antakelser med tanke på oppbygningen av brukte emnekart, for å ha et fungerende program tilgjengelig relativt tidlig og innenfor rammene av tildelte ressurser. Foreløpig støttes kun binære relasjoner, men ettersom relasjoner av høyere aritet kan brytes ned til 2-plass relasjoner er ikke dette noen alvorlig mangel. Å anta at alle relasjoner er binære forenkler i stor grad den interne oppbygningen av emnekartrepresentasjoner i PyToM, og gjør det



Figur 4.3: Fletting av emnekart (ref. Figur 4.2) inneholdende emner med sammenfallende subjektidentitet.

lettere å forsikre seg om at ens informasjonsmodeller er konsistente og uten feil med hensyn til assosiasjoner. For PyToM gjelder det dessuten at relasjoner er implementert i henhold til mengdelæren, hvor en binær relasjon er definert som en mengde ordnede par (ref. avsnitt 3.2.1). Dette er gjort ved at en 'PyToM-kompatibel' XTM-relasjon må inneholde to assosiasjonsroller (member-elementer), som igjen må inneholde det samme antallet emnereferanser. Elementene fra hver av assosiasjonsrollene 'pares' nemlig koordinatvis, og to emner med samme kooridinat utgjør dermed et par i den binære relasjonen. Emner med samme koordinat i første og andre assosiasjonsrolle utgjør med andre ord et ordnet par i PyToM: 1. element av 1. assosiasjonsrolle pares med 1. element av 2. assosiasjonsrolle, 2. element av 1. assosiasjonsrolle pares med 2. element av 2. assosiasjonsrolle, ..., n. element av 1. assosiasjonsrolle pares med n. element av 2. assosiasjonsrolle (se appendiks A for kommentert utdrag av XTM).

Referanser til hver enkelt relasjon lagres deretter i en tabell av relasjoner i en av klassene i PyToM. Denne tabellen reflekterer den koordinatvise fordelingen av parene i hver enkelt relasjon, og en representasjon av at emnet identifisert ved ID redskap deltar i relasjoner av typen superclass-subclass_association med emnene steinredskap, treredskap og beinredskap, hvor også emnene steinredskap og flint deltar i samme type relasjon, vil se noenlunde ut som vist nedenfor:

{ superclass-subclass_association: { redskap: [steinredskap, treredskap, beinredskap], steinredskap: [flint] } }

At assosiasjoner lagres på denne måten gjør det for eksempel til en relativt triviell oppgave å generere den transitive lukningen av en relasjon. For å gå fra 'toppen' til 'bunnen' i relasjonen ovenfor beveger en seg fra redskap til flint, og en trenger altså bare hente ut siste element i tabellen av redskaps relaterte emner. Videre åpner det for at en enkelt kan finne inversen av en relasjon, for eksempel av relasjonen mellom flint og steinredskap; man lar bare nøkler og verdier bytte plass i listen. Dette benyttes blant annet ved perspektivering av assosiasjoner i PyToM, hvor fastslåing av en gitt relasjons invers muliggjør visning av relasjonen fra begge 'sider'. En kan for eksempel presentere brukere for et funn som er $funnetved$ en heller, mens helleren beskrives som $funnplassfor$ det gitte funnet (og ikke $funnetved$ funnet).

4.2.3 Vevapplikasjon. Navigering og søk

For å aksessere emnekartrepresentasjoner i PyToM benytter vevapplikasjonen utviklet for Hellerprosjektet en samling egne Python-klasser. Disse klassene representerer nettsider og er bygget med utgangspunkt i en felles klasse (SimpleWebPage), som er spesialisert etter behov (SimpleWebPage → TopicMapPage → TopicPage, etc.). En klasse for emnekartsider inneholder her metoder som kan kalle emnekart-spesifikke metoder i PyToM, en klasse for emnesider inneholder metoder for emne-spesifikke operasjoner i PyToM, og så videre. Disse Python-klassene benyttes så av Spyce-skript[8] som utgjør den delen av applikasjonen som er synlig for, og kalles av, klienter (for eksempel nettlesere). Spyce-skriptene oppretter objekter og jobber mot Python-skriptene, sender informasjon til egne templater som inneholder presentasjonslogikk (HTML + Spyce kode), og skriver ut HTML-kode som sendes fra server til klient. På denne måten omdannes det i en dynamisk prosess HTML av XTM, og en er sikret at HTML-koden til enhver tid reflekterer informasjonen og -typene representert i XTM-dokumentene.

Foreløpig er bare mindre deler av nettstedet bygget opp med utgangspunkt i emnekart. Dette skyldes i all hovedsak at Hellerprosjektet fra tidlig av hadde et sterkt behov for å få nettstedet 'på lufta'. I utgangspunktet ble det derfor opprettet et nettsted med funksjoner for å blant annet publisere artikler. Dette nettstedet er så etter hvert

[8] "Python Server Pages"; `http://www.spyce.org/`.

blitt videreutviklet. I tillegg gjelder det at selve ontologiutviklingen - med tilhørende XTM-koding - utføres av mastergradsstudent Jan Erik Mandelid ved Seksjon for humanistisk informatikk. For denne oppgavens vedkommende har det derfor vært viktigere å skape et rammeverk for web-formidling, enn å duplisere Mandelids arbeid. Det er dog benyttet midlertidige emnekart på enkelte deler av nettstedet. Formålet med disse delene av nettstedet er å eksemplifisere og utforske bruk av assosiative informasjonssturkturer for navigering i forskningsmaterialet.

Navigering

I nettstedets meny fører to valg frem til ulike emnekart: 'Hellere på Vestlandet' → 'Bilder av hellere' (`http://huin.uib.no/hellerprosjektet/images.spy`) og 'Prosjekt-emnekart' (`http://huin.uib.no/hellerprosjektet/showtopicmaps.spy`). Under 'Prosjekt-emnekart' listes alle emnekart som befinner seg i applikasjonens XTM-beholdning. I skrivende stund er disse å anse for test-data, men inneholder allikevel data som illustererer navigeringsfunksjonene utviklet så langt. Dersom en velger å utforske et spesifikt emnekart, vil en få opp en problemspesifikk taksonomi (inneholder ikke 'husholdnings-artikler' som språk, forekomst-typer, etc.) over klasser i emnekartet. Den problemspesifikke taksonomien må være eksplisitt angitt, slik at Python-skriptene kan avgjøre hvilke emner som utgjør den problemspesifikke delen av emnekartet. Figur 4.4 viser et eksempel på en taksonomi vist i vevapplikasjonen.

Figur 4.4: "Innsnevret" problem-spesifikk taksonomi.

I de dynamisk genererte HTML-dokumentene som gjengir taksonomier listes alle emneklasser, og ved å klikke på en klasse beveger en seg til en ny side (showtopic.spy) som viser informasjon relatert til den gitte klassen - for eksempel ulike navn, instanser, foreldreemner og subjektidentitet. Figur 4.5 viser et eksempel på en HTML-side over emnet "Vasselhelleren" (en instans av klassen "Hellere i Herand" fra Figur 4.4). Denne nye siden viser eventuelle forekomster, relasjoner og foreldreemner av valgte emne, samt linker til ressurser som utgjør eller indikerer emnets subjektidentitet. Vi ser her hvordan

assosiativiteten i emnekartet reflekteres i relasjonslinkene vist på HTML-siden, og ved å klikke på linken i relasjonen "Vasselhelleren graves ut av Hellerprosjektet" beveger en seg til (det relaterte) emnet "Hellerprosjektet" (Figur 4.6). Også her listes relevant informasjon, og vi ser blant annet at relasjonen *graves ut av* fra Figur 4.5 nå vises fra perspektivet *graver ut*. Fra "Hellerprosjektet" kan en så bevege seg videre i informasjonsmodellen, blant annet til emnet "Knut Andreas Bergsvik" som viser informasjon relatert til Hellerprosjektets prosjektleder. Fra "Knut Andreas Bergsvik" kan en igjen bevege seg til for eksempel emnet "Arkeolog", fra "Arkeolog" til "Haakon Shetelig", fra "Haakon Shetelig" til "Ruskenesset", fra "Ruskenesset" til "Bergen Museum", fra "Bergen Museum" til "Grønehelleren", fra "Grønehelleren" til "Skjelettet fra Grønehelleren", og så videre.

Figur 4.5: Visning av emnet "Vasselheller" i en nettleser. Svakt markert område under "Relasjoner" viser linken som klikkes på for å gå til emnet "Hellerprosjektet", vist i Figur 4.6.

Figur 4.6: Visning av emnet "Hellerprosjektet" i en nettleser.

Ut fra figurene 4.4 - 4.6 ser vi hvordan en gjennom selvbeskrivende ressurser kan gjøre informasjon tilgjengelig for ulike typer applikasjoner. Typer, relasjoner og andre former for informasjon ligger her lagret i informasjonsrike dokumenter som i utgangpunktet kan tolkes av enhver XML parser. Vidt forskjellige applikasjoner kan derfor utformes for å benytte seg av den informasjonen som en nedfeller i standardiserte formater som XTM. Dette gjenspeiles også i Figur 4.1 hvor det ikke er Hellerprosjektets applikasjon som presenterer data fra et emnekart, men programmet TMNav. Relasjoner, med mer, er ikke hardkodet på applikasjons-, men på dokument-nivå, hvilket er hvorfor TMNav er i stand til å presentere ulike informasjonstyper slik de er beskrevet av det enkelte emnekarts forfatter. Ved at objekter i et diskursdomene er spesifisert i ontologier kan en altså eksplisitt beskrive informasjonmodeller hvor ulike informasjonsbiter står i et innbyrdes forhold til hverandre. En kan derfor utforske informasjonsmengden basert på den assosiativiteten som 'naturlig' befinner seg i et diskursdomene (i tråd med Bushs tanker, ref. avsnitt 2.2)

- vel og merke sett fra forfatterens perspektiv. Samtidig er XTM-dokumentene i seg selv informasjonsressurser, og ulike systemer kan derfor uttrykke påstander om disse og slik utvide informasjonsmodellene ut fra egne behov. Slik kan betydningsbevarende informasjonsstrukturer som emnekart gjøre mennesker, så vel som maskiner, i stand til å utforske og gjenfinne informasjon basert på egenskaper ved informasjonen selv.

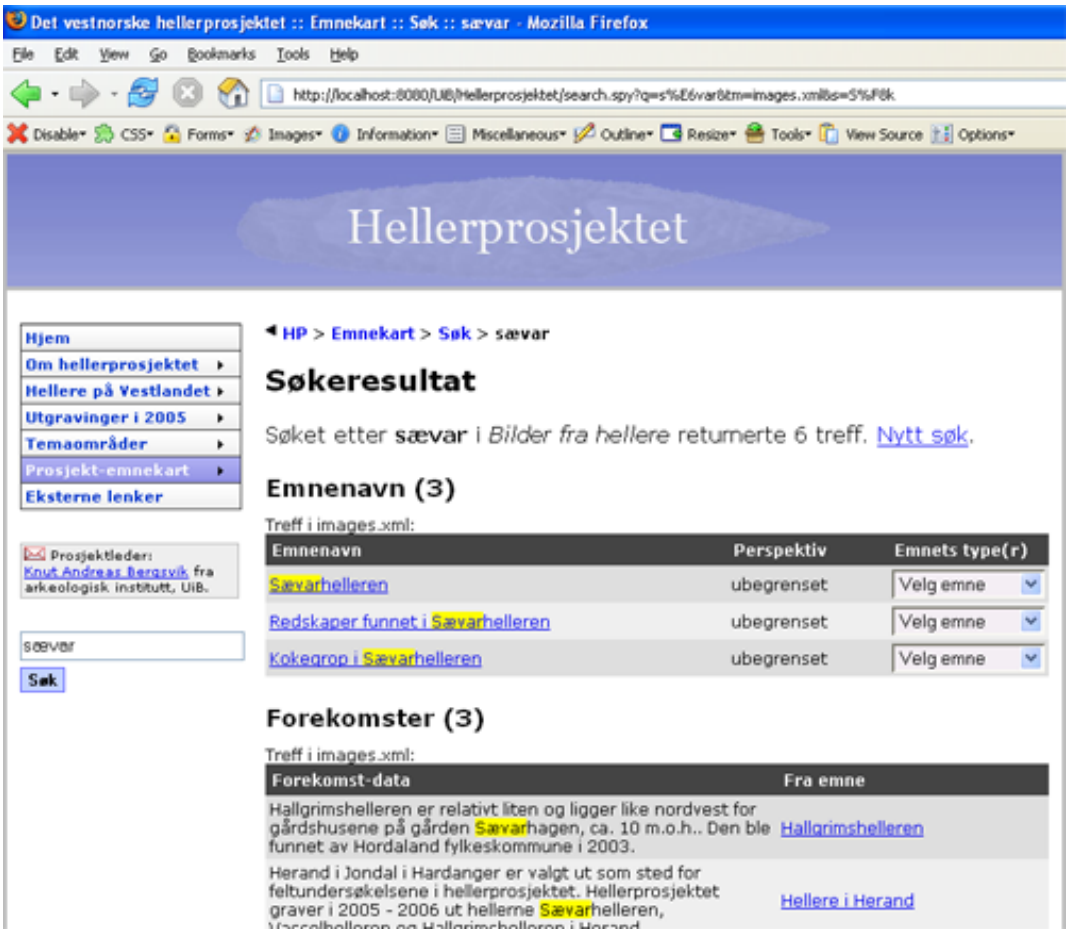
Søk i XTM

For at det skal være mulig å finne informasjon om emner uten å måtte bla seg gjennom hele informasjonsmodellen, har det som en del av arbeidet med Hellerprosjektets vevapplikasjon blitt utviklet en intern søkefunksjon (`http://huin.uib.no/hellerprosjektet/search.spy`). Denne søkefunksjonen søker etter en tekststreng (søkefrase gitt av bruker) som en del av samtlige emnenavn og forekomster i valgte XTM-fil, eventuelt alle XTM-filer. Eventuelle resultater fra slike søk ordnes etter XTM-fil, og presenteres i tabeller over emnenavn og forekomster som inneholder søkestrengen. I tabellene over emnenavn presenteres også navnets gyldighetsområde (perspektivet) og eventuelle typer - dersom emnet er en instans. For treff i forekomster presenteres selve forekomsten, samt emnets navn (under 'unconstrained scope'). For begge typene treff gjelder det at emnenavn (og type) er klikkbare linker (Figur 4.7), slik at en kan benytte disse for videre utforsking av informasjonsmengden. Denne ordningen av søkeresultater er i stor grad inspirert av Garshol (2004).

Selv om dette foreløpig er et relativt 'naivt' søk, viser det hvordan betydningsorientert merking har et potensiale også innenfor området Information Retrieval. Ved å legge til muligheter for å avgrense søket til å gjelde bare en eller flere bestemte typer, noe som foreløpig ikke er implementert i vevapplikasjonens søkefunksjon, hadde en for eksempel kunnet konstruere dataprogrammer som i større grad kunne funnet relevante treff, enn hva som er tilfellet for rene fulltekstsøk (ref. avsnitt 3.1). At søkeresultater presenteres ut fra ressursenes typer kan naturligvis også bidra til at en menneskelig bruker lettere kan avgjøre hvorvidt et gitt søketreff faktisk er relevant.

Vevapplikasjonen støtter foreløpig bare fulltekstsøk i XTM-filer, men i assosiative informasjonsinfrastrukturer kan en også tenke seg mer avanserte søk som for eksempel benytter mengdeteoretiske operasjoner som union, snitt og komplement for å finne instanser av bestemte typer emner. I kombinasjon med et fulltekstsøk vil dette kunne være en svært effektiv form for søk (med tanke på relevans), men selv om utviklingen av et slikt søk for Hellerprosjektet ble påbegynt (`http://huin.uib.no/hellerprosjektet/`

`asearch.spy`) i forbindelse med dette mastergradsprosjektet, har det dessverre ikke latt seg gjøre å ferdigstille Web-grensesnittet for et slikt søk innenfor gjeldende tidsrammer. Mulighetene for dette er dog innebygget i PyToM, og har vært eksperimentert med (utenfor vevapplikasjonen) i løpet av utviklingen.



Figur 4.7: Søk etter 'sævar' i 'Bilder fra hellere' (images.xml).

4.2.4 Forslag til videre utvikling

I et prosjekt som dette vil det nesten alltid være rom for forbedringer hva gjelder det praktiske arbeidet, og Hellerprosjektets vevapplikasjon er intet unntak. Som nevnt i avsnitt 4.2.3 hadde det eksempelvis vært ønskelig å implementere et avansert søk som benyttet seg av PyToMs muligheter for mengdeoperasjoner, men grunnet tidsbegrensninger

lot dette seg ikke fullføre. Ettersom det fra relativt tidlig av var ønskelig å få nettstedet on-line, og det med jevne mellomrom dukket opp forespørsler om nye elementer som måtte legges til, har det ikke alltid vært mulig å prioritere det praktiske arbeidet relatert til rapportens teoretiske fundament. Selv om dette er en naturlig prosess innenfor web-utvikling, har det resultert i at enkelte deler av vevapplikasjonen med fordel kan forbedres. Det gjøres derfor her kort rede for mulig videreutvikling og potensielle forbedringer, slik forfatteren av vevapplikasjonen ser det.

Ontologi

Den foreløpige ontologien utviklet for vevapplikasjonen er som tidligere nevnt bare å anse for test-, eller 'dummy'-data. Dette prosjektet har ikke hatt som oppgave å utvikle en ontologi, og det har heller ikke vært ønskelig å utvikle en ontologi parallelt med utviklingen av 'hoved-ontologien' ved Jan Erik Mandelid. Uansett hvordan ontologien som er representert i foreløpige emnekart er designet, så har den, gjennom å bidra til å belyse oppgavens problemstilling, oppfylt sitt formål. Med det sagt finnes det åpenbare svakheter i den midlertidige ontologien / midlertidige emnekart. Bare enkelte av objektene fra en arkeologi- eller hellerontologi er representert i emnekartene, og ettersom emnekartenes funksjon er å være test-data, vil en tilførsel av nye konsepter lett 'ødelegge' ontologien. Den foreløpige ontologien lider altså av slett design, hvilket vil si at det eksisterer et behov for bedre ontologier og en mer gjennomført inndeling av blant annet funn fra hellerne. Som tidligere nevnt er dette underveis.

For de klasser som publiseres som en del av hellerprosjektet, og som det ikke allerede finnes publiserte subjekter og subjektindikatorer for - for eksempel subjektet 'heller', anbefales det at Hellerprosjektet forsøker å gjøre PSIer offentlig tilgjengelig. Dette for å støtte opp om gjenbruk av ontologier, og dermed også visjonen om en semantisk verdensvev og XTM-"miljøet", hvis kunnskap prosjektet har nytt godt av.

Øvrig kode

Deler av Python- og Spyce-koden i vevapplikasjonen vitner om at mange ulike oppgaver har vært forsøkt løst i løpet av forholdsvis kort tid. Som innledningsvis nevnt har det ikke blitt lagt til filtrering basert på perspektivering (`scope`) i XTM. Grunnet prioritering av andre funksjoner, samt det faktum at denne siden ved XTM i og for seg ikke har vært viktig for oppgavens diskusjon, støtter vapplikasjonen per dags dato ikke filtrering basert på

XTMs scope. Filtrering er dog tilgjengelig i PyToM (gjennom metoder som for eksempel PyToM.TypedElements.Topic.get_occurrences_in_scope()), og kan forholdsvis enkelt legges til vevapplikasjonen. For Spyce-koden og Spyce/HTML-templatenes vedkommende er filtreringsmekanismen nemlig likegyldig, og det vil således bare være nødvendig å gjøre endringer i Python-filer for at filtrering av informasjon skal kunne reflekteres på nettstedet.

Ettersom filtrering ikke er implementert, er det heller ikke mulig å tilpasse nettstedets sider for ulike klienter - for eksempel mindre informasjon per side for mobiltelefoner versus PCer, basert på informasjon nedfelt i XTM-filene. I stedet benyttes ulike templater for ulike klient-typer. Foreløpig er bare HTML-templater lagt til nettstedet, men dersom nettstedets templatmappe (/includes/templates) inneholder en katalog med for eksempel WML (Wireless Markup Language)-filer vil disse velges over HTML (for WML-støttende klienter). Slike templater må, i likhet med eksisterende HTML-templater, inneholde presentasjonslogikk for å begrense utskrevet informasjonsmengde, da informasjonen som sagt ikke vil filtreres med bakgrunn i perspektivering i XTM-filer. Et Python-skript (UserAgent.py) avgjør hvorvidt klienten synes å foretrekke for eksempel WML, og kontrollerer deretter om denne templattypen eksisterer. Hvis templatkontrollen feiler vil HTML velges som standard templat-type, hvilket er hva som skjer i dag. For språk gjelder det at alle tekststrenger hentes ut fra egne språkpakker og sendes til de ulike templater i form av en liste.

For andre sider av nettstedet gjelder det også at koden ikke er 100% optimalisert. Blant annet er fulltekstsøket som tidligere nevnt relativt 'naivt'. Denne, og andre funksjoner, benytter for eksempel gjentakende nøsting av for-løkker for iterering over lister av data, noe som kan innebære at enkelte algoritmer blir tidkrevende dersom informasjonsmengden øker. Selv om det antas at dette ikke vil utgjøre et problem for Hellerprosjektets vevapplikasjon, kan det altså med fordel implementeres bedre algoritmer i enkelte deler av vevapplikasjonen. Videre gjelder det at objektorientering i applikasjonen muligens kunne vært forbedret, samt at en for bedre brukervennlighet med tanke på administrering av nettstedet med fordel kunne flyttet større deler av Spyce-koden fra Spyce-templater til Spyce- og/eller Python-skript, og eventuelt kapslet lister av data inn i klasser. Dette for å minimere innholdet av presentasjonslogikk i Spyce-/HTML-templater, som ideelt sett bør inneholde så lite Spyce-kode som mulig slik at en endring av templater ikke nødvendigvis forutsetter kjennskap til Spyce/Python. Deler av denne problematikken er dog forsøkt løst gjennom at all layout-spesifikk informasjon er lagret

i stilark (CSS), ettersom en ved hjelp av CSS kan styre så å si alt hva gjelder layout og design.

Grensesnitt

Et ytterligere aspekt ved vevapplikasjonen som ikke har vært prioritert, er utvikling av grensesnitt og design. Det er forsøkt utformet et simplistisk grensesnitt i henhold til Jacob Nielsens (2000) berømte mantra for godt web-design, hvor informasjon har forrang over for eksempel grafikk og farger. For eksempel består hver side av tre hoveddeler (topp, meny og kropp), tekstlinker indikeres gjennom (standard) understreket, blå tekst, og brukere vil alltid informeres om hvor de er (toppbanner og 'navigasjonsmeny' øverst på hver side). Til tross for dette gjenstår fremdeles enkelte elementer med tanke på vevapplikasjonens design og navigasjonsmuligheter. Sannsynligvis er for eksempel språkbruken ikke egnet for slutt-brukere (ord som 'Taksonomi', 'Forekomster', 'Relasjoner', etc.). Videre kan det tenkes at brukere ønsker å kunne gå fra et emne til dets taksonomiske søsken. Det vil si å kunne bevege seg mellom for eksempel 'Hellere i Herand' til 'Tidligere utgravde hellere' (Figur 4.4), eller fra en tenkt 'Flintkniv #1' til 'Flintkniv #2'. Selv om denne informasjonen er nedfelt i emnekartene, er en slik navigasjonsmulighet per i dag ikke implementert i Hellerprosjektets applikasjon. En kan riktignok bevege seg til søskenemner ved å gå via eventuelle foreldreemner, men dette kan selvsagt ses på som en omvei / et unødvendig steg (krever 2 ekstra museklikk, foruten at brukeren må være klar over muligheten).

4.3 Oppsummerende kommentarer

Vi har nå sett hvordan lærdom hentet fra distribuert prosessering og kunstig intelligens kan sies å ligge til grunn for en betydningsorientert verdensvev, og hvordan denne kunnskapen er forsøkt utnyttet i praksis. For Hellerprosjektets vedkommende var XML Topic Maps rette teknologi å ta i bruk som et publiseringsverktøy, for andre prosjekter vil gjerne andre semantiske netteknologier synes bedre. Som alltid finnes det fordeler og ulemper når ulike løsninger veies opp mot ens målsetninger, og avslutningsvis gjøres det her noen oppsummerende tanker omkring betydningsorientert forskningsformidling på Web - blant annet for å rette søkelyset mot andre problemstillinger som kan reises i tilknytning til kunnskapsrepresentasjon og forskningsformidling.

4.3.1 Betydningsorientert forskningsformidling på Web

Denne masteroppgaven har vist hvordan en betydningsmerket informasjonsbase kan danne grunnlag for forskningsformidling - også innenfor humaniora. Gjennom å beskrive deler av verden i ontologier, og gjøre ontologier tilgjengelig for datamaskinell prosessering ved hjelp av standardiserte formater, muliggjør en informasjonsutveksling og -deling mellom heterogene systemer. Ved å benytte utvetydige identitetskriterier kan en videre forsikre seg om at objektene i ens ontologier er entydig spesifisert. Autoklassifiserende informasjonstrukturer gjør det dessuten relativt lett å tilføye nye elementer til en informasjonsmodell, ettersom en simpelthen kan identifisere elementets plassering i strukturen. Deretter vil elementet gjennom subsumering arve egenskaper fra samtlige av dets superklasser.

Ved hjelp av semantiske netteknologier kan informasjonsmodeller som sagt eksplifiseres i et standardisert, maskinleselig format og gjøres tilgjengelig for maskinell prosessering, gjerne på tvers av applikasjoner i et distribuert nettverk. Slike teknologier åpner dermed for deling av informasjon merket på en måte som til en viss grad kan sies å være kunnskapsbevarende[9], noe som bør appellere til både humaniora og akademia generelt. I assosiative informasjonstrukturer kan en videre strukturere informasjon i oversiktlige taksonomier, samtidig som enkeltelementer bevares fragmentert - noe som muliggjør assosiativ navigering mellom ulike enheter. For humanistisk forskning gjelder det altså at en kan presentere allerede tolket informasjon på andre måter enn rent narrativt[10], og dermed kanskje legge opp til andre former for møter mellom brukere og fortolket materiale. Dersom et prosjekt som Hellerprosjektet publiserer deler av sine resultater i formater som OWL eller XTM vil en dessuten i andre applikasjoner kunne benytte seg av informasjonsressursene (slik de er modellert av Hellerprosjektet), samtidig som en i godt modellerte informasjonsmengder maskinelt kan avgjøre betydningen til enkeltelementer. En kan dessuten publisere PSIer for modellerte klasser og typer for å forsøke å eliminere problemer knyttet til tvetydighet, slik at ulike systemer kan oppnå enighet med hensyn til betydning. Dersom Bergen Museum ønsker å lage en nettpresentasjon av deler av Hellerprosjektets materiale kan en eksempelvis lage egne abstraksjoner av informasjonen lagret i prosjektets XTM-dokumenter og/eller flette egne informasjonsmodeller med Hellerprosjektets, og presentere disse på måter tilpasset museets egne brukere. Eventuelt

[9] Gitt at kunnskapsbevaring i effekt betyr informasjons-, identitets- og betydningsbevaring for de konsepter som forfatteren av en informasjonsmodell finner vesentlige.

[10] I den grad 'vanlige' hypertekster kan regnes som narrative.

kan en ved hjelp av Hellerprosjektets publiserte subjektindikatorer, dersom slike foreligger, reifisere Hellerprosjektets publiserte subjekter, og på denne måten sikre seg mot tvetydighet i informasjon hos adskilte nettsteder.

Økning av data - økning av kunnskap?

At en kan flette informasjon hentet fra ulike dokumentbaser er et viktig element i semantiske netteknologier, og fletting av uavhengige indekser var også - som nevnt i avsnitt 4.1 - et av målene bak utviklingen av emnekartteknologien. Gjennom fletting utnytter en informasjon fra eksisterende ressurser, og fjerner identifiserbar overflødighet. På grunn av at en slik kan skape en virtuell samlokalisering av ulike informasjonsmodeller, kan en gjerne ledes til å tro at fletting løser problemer knyttet til mikroverdener (ref. avsnitt 2.3.1). Gitt at identitetskriteriene benyttet i ulike ontologier er de samme kan det for eksempel være fristende å tro at en kan utvide mikroverdener simpelthen gjennom å flette en ontologi med en annen ontologi, som kan flettes med en tredje ontologi, som igjen kan flettes med en fjerde ontologi, og så videre. Som tidligere nevnt (avsnitt 3.2) er dog ikke dette tilfellet; ontologier designes ofte som 'spesialiserte' mikroverdener, og på et gitt detaljnivå (Sowa 2000). Derfor vil også fletting av kunnskapsbaser være problematisk. Ulike forfattere kan dessuten benytte samme subjektidentitet (inkludert PSIer i emnekart) for vidt forskjellige subjekter. Forfatterens tolkning av, og intensjoner med, et diskursdomene spiller altså en vesentlig rolle, og kan derfor føre til at en ved fletting av ontologier står igjen med unøyaktig eller uriktig informasjon. I en slik prosess kan det for eksempel hende at ens emner "plutselig" er direkte (uten 'mellomledd') assosierte med emner som, ut fra ens eget perspektiv, absolutt ikke burde være assosierte. En kan derfor kanskje hevde at ved fletting kan riktignok informasjons-, eller datamengden, øke, men også at den egentlige *kunnskapen* kan synke? Riktignok kan dette problemet delvis løses gjennom bruk av tillit og sertifiserte tredjeparter, men det forblir likefullt et problem etter hvert som informasjonsmengdene øker og prosesser som fletting i stadig større grad automatiseres. For forskningsformidling er dette selvsagt særlig relevant, idet en her vil være spesielt interessert i å formidle korrekt og sannferdig informasjon. Foreløpig er det derfor mest sannsynlig at en 'ekte' semantisk verdensvev - med sine autonome agenter - også i nærmeste fremtid i all hovedsak vil være en visjon, og at en vanskelig kan bevege seg utenfor mikroverdener eller isolerte systemer hvor en (til en viss grad) kan kontrollere informasjonsflyten og -mengden.

Samtidig gjelder det at nyere semantiske netteknologier som emnekart utvilsomt gir

oss nye muligheter, også innenfor områder som forskningsformidling og E-læring (Rittershofer 2005; Jopp 2004). Hellerprosjektets nettsted vil, som beskrevet i denne rapporten, danne grunnlag for formidling av arkeologisk forskningsmateriale. Ettersom humanistisk forskningsmateriale i utgangspunktet er fortolket kan en assosiativ informasjonstruktur kanskje i særdeleshet sies å være et interessant verktøy for formidling av forskningsresultater innen humaniora. Assosiative informasjonsinfrastrukturer tillater jo nettopp utforsking av informasjonsmodeller sett fra forfatters perspektiv. Dette elementet kan selvsagt også utnyttes i spesifikke læringssituasjoner - hvor elever for eksempel selv kan uttrykke sammenhenger, men selv om en ut fra beskrivelser som i Park & Hunting (2003) kan komme til å tro at emnekart nærmest er en reddende engel for undervisning og læring, er også semantiske netteknologier i denne sammenheng bare for verkøy å regne. Ofte kan det derfor være nyttig å minnes på at teknologier utvikles av teknologer, og ikke nødvendigvis i samarbeid med pedagoger[11].

[11]Se blant annet Tore Hoels (2002) diskusjon omkring standardisering av E-læring generelt, *Bygges morgendagens læringsteknologi på gårsdagens læringssyn?*, for mer om dette.

Tillegg A

Kildekode

A.1 Utdrag fra midlertidig XTM

Under følger et utdrag fra emnekartet images.xml, brukt i eksempler. Legg merke til hvordan assosiasjoner er modellert som binære relasjoner (mengde ordende par), samt hvordan forekomster reifiseres for å relatere tekst til bilder. Hele emnekartet er tilgjengelig på `http://huin.uib.no/hellerprosjektet/xtm/images.xml`, samt på vedlagte CD-ROM. Merk at dette emnekartet bare er brukt som test- / "dummy"-data for vevapplikasjonen beskrevet i oppgavens kapittel 4.

```
<topicMap xmlns="http://www.topicmaps.org/xtm/1.0/"
          xmlns:xlink="http://www.w3.org/1999/xlink" id="images_xtm">

  <!-- Reifikasjon av emnekart  -->
  <topic id="images_xtm_reif">
    <subjectIdentity>
      <subjectIndicatorRef xlink:href="#images_xtm" />
    </subjectIdentity>
    <baseName>
      <baseNameString>Bilder fra hellere</baseNameString>
    </baseName>
    <occurrence>
      <topicRef xlink:href="#decription" />
      <resourceData>
        Bilder-XTM.
        Et `forminsket' eksempel av emnekartet brukt
        for eksemplene i oppgaven.
```

```
    </resourceData>
  </occurrence>
</topic>
<!-- / Reifikasjon av emnekart  -->

<!-- Husholdningsartikler  -->

<topic id="hellerprosjekttaxon">
  <baseName>
    <baseNameString>Univers (alle emner)</baseNameString>
  </baseName>
</topic>
<topic id="image">
  <baseName>
    <baseNameString>Bilde</baseNameString>
  </baseName>
</topic>
<topic id="thumbnail">
  <baseName>
    <baseNameString>
      Tommelnegl / miniatyr-versjon av bilde
    </baseNameString>
  </baseName>
</topic>
<!-- ytterligere emner -->

<!-- Problem-spesifikke emner -->

<!-- Klasser -->
<topic id="heller">
  <subjectIdentity>
    <subjectIndicatorRef
      xlink:href="http://huin.uib.no/hellerprosjektet/tema.spy" />
    <subjectIndicatorRef
      xlink:href="http://no.wikipedia.org/heller" />
  </subjectIdentity>
     <baseName>
       <baseNameString>Heller</baseNameString>
     </baseName>
     <occurrence>
        <instanceOf>
            <topicRef xlink:href="#text" />
         </instanceOf>
         <resourceData>Hellere er overheng av berg.</resourceData>
  </occurrence>
</topic>

<topic id="hpheller">
  <subjectIdentity>
    <!-- Hellere som graves ut av Hellerprosjektet -->
    <subjectIndicatorRef
      xlink:href="http://huin.uib.no/hellerprosjektet/2005.spy" />
  </subjectIdentity>
  <baseName>
    <baseNameString>Hellere i Herand</baseNameString>
  </baseName>
```

```
  <occurrence>
    <instanceOf>
      <topicRef xlink:href="#text" />
    </instanceOf>
    <resourceData>
      Herand i Jondal i Hardanger er valgt ut....
    </resourceData>
  </occurrence>
</topic>

<topic id="funn">
  <baseName>
    <baseNameString>Funn</baseNameString>
  </baseName>
</topic>

<topic id="redskap">
  <baseName>
    <baseNameString>Redskap</baseNameString>
  </baseName>
</topic>

<topic id="fiskekrok">
  <baseName>
    <baseNameString>Fiskekrok</baseNameString>
  </baseName>
</topic>

<topic id="fiskestikke">
  <baseName>
    <baseNameString>Fiskestikke</baseNameString>
  </baseName>
</topic>

<topic id="beinredskap">
  <baseName>
    <baseNameString>Beinredskap</baseNameString>
  </baseName>
</topic>

<!-- ytterligere emner -->

<!-- Assosiasjons-typer -->

<topic id="funnplass">
  <baseName>
    <baseNameString>Funnplass</baseNameString>
  </baseName>
</topic>

<topic id="funnet_ved_assoc">
  <baseName>
    <baseNameString>Funnet-ved-relasjon</baseNameString>
  </baseName>
  <baseName>
    <scope><topicRef xlink:href="#funnplass" /></scope>
```

```
    <baseNameString>er funnplass for</baseNameString>
  </baseName>
  <baseName>
    <scope><topicRef xlink:href="#funn" /></scope>
    <baseNameString>ble funnet ved</baseNameString>
  </baseName>
</topic>

<!-- Instanser -->

<!-- hellere -->
<topic id="saevarhelleren">
  <subjectIdentity>
    <subjectIndicatorRef
    xlink:href="http://huin.uib.no/hellerprosjektet/news.spy?newsid=10" />
  </subjectIdentity>
  <instanceOf>
    <topicRef xlink:href="#hpheller" />
  </instanceOf>
  <baseName>
    <baseNameString><![CDATA[ S&aelig;varhelleren ]]></baseNameString>
  </baseName>
  <occurrence>
    <instanceOf>
      <topicRef xlink:href="#text" />
    </instanceOf>
    <resourceData>
      <![CDATA[ S&aelig;varhelleren ligger p&aring; g&aring;rden... ]]>
    </resourceData>
  </occurrence>
  <occurrence id="DSC_0145">
    <instanceOf>
      <topicRef xlink:href="#thumbnail" />
    </instanceOf>
    <resourceRef
      xlink:href="images/hellere/saevarhelleren/DSC_0145-sml.jpg" />
  </occurrence>
  <!-- ytterligere forekomster -->
</topic>

<!-- øvrige hellere -->

<!-- funn -->
<topic id="fiskeutstyr_020106">
  <instanceOf>
    <topicRef xlink:href="#fiskekrok" />
  </instanceOf>
  <instanceOf>
    <topicRef xlink:href="#fiskestikke" />
  </instanceOf>
  <instanceOf>
    <topicRef xlink:href="#beinredskap" />
  </instanceOf>
  <baseName>
    <baseNameString>Fiskeutstyr av bein</baseNameString>
  </baseName>
```

```
  <occurrence id="020106_12">
    <instanceOf>
      <topicRef xlink:href="#thumbnail" />
    </instanceOf>
    <resourceRef
      xlink:href="images/hellere/saevarhelleren/DSCN0986-sml.jpg" />
  </occurrence>
</topic>

<!-- ytterligere funn -->

<!-- Reifiserte forekomster -->
<topic id="020106_reif_a8">
  <subjectIdentity>
    <!-- Fiskeutstyr av beins thumbnail-forekomst -->
    <subjectIndicatorRef xlink:href="#020106_12" />
  </subjectIdentity>
  <occurrence>
    <!-- Assosier tekst-info. med bildet -->
    <instanceOf>
      <topicRef xlink:href="#text" />
    </instanceOf>
    <resourceData>
      <![CDATA[
      Fiskekroker og fiskestikke av bein fra S&aelig;varhelleren.
      9000 &aring;r gamle. Fiskestikka er ca. 5 cm lang.
      ]]>
    </resourceData>
  </occurrence>
</topic>

<!-- ytterligere emner og reifikasjoner -->

<!-- Assosiasjoner -->

<!-- Taksonomi (kun subsumering) -->
<!-- emner fra 1. member og 2. member pares koordinatvis -->
<association id="superclass-subclass_association">
  <instanceOf>
    <subjectIndicatorRef
    xlink:href="http://www.topicmaps.org/xtm/1.0/core.xtm#superclass-subclass"/>
  </instanceOf>
  <member>
    <!-- super-forelder -->
    <topicRef xlink:href="#hellerprosjekttaxon" />
    <topicRef xlink:href="#hellerprosjekttaxon" />

    <!-- forelder av ulike hellertyper -->
    <topicRef xlink:href="#heller" />

    <!-- forelder av ulike funntyper -->
    <topicRef xlink:href="#funn" />

    <!-- ytterligere emne-referanser -->
  </member>
  <member>
```

```
      <!-- barn av hellerprosjekttaxon -->
      <topicRef xlink:href="#funn" />
      <topicRef xlink:href="#heller" />

      <!-- barn av heller -->
      <topicRef xlink:href="#hpheller" />

      <!-- barn av funn -->
      <topicRef xlink:href="#redskap" />

      <!-- ytterligere emne-referanser -->
    </member>
  </association>

  <!-- Øvrige assosiasjoner -->

  <association id="funnet_ved_assoc_020106">
    <instanceOf>
      <topicRef xlink:href="#funnet_ved_assoc" />
    </instanceOf>
    <member>
      <roleSpec>
        <topicRef xlink:href="#funnplass" />
      </roleSpec>
      <!-- funnsteder / hellere -->
      <topicRef xlink:href="#saevarhelleren" />
      <!-- ytterligere emne-referanser -->
    </member>
    <member>
      <roleSpec>
        <topicRef xlink:href="#funn" />
      </roleSpec>
      <!-- fiskeutstyr_020106 er funnet ved saevarhelleren -->
      <topicRef xlink:href="#fiskeutstyr_020106" />
      <!-- ytterligere emne-referanser -->
    </member>
  </association>

  <!-- ytterliggere assosiasjoner og emner -->

</topicMap>
```

A.2 Python- / Spyce-kode

Se vedlagte CD-ROM for filer.

Tabellen under viser fil- og mappestrukturen på vedlagte CD-ROM.

Fil- / mappenavn	Beskrivelse
/PyToM	PyToM-filer
/flash	Flash .fla fil for "Hellere på Vestlandet"
/images	Bildemappe. Bilder fra hellere (følger ikke med på CD-ROM grunnet opphavsrettigheter), ikoner, etc.
/includes/cfg/config.py	Konfigurasjonsfil. Valg for antall artikler per side, etc.
/includes/classes	Alle klasser brukt av Spyce skript
/includes/lang_packs	Alle språkpakker. Se /includes/lang_packs/README.txt for info.
/includes/templates/html	HTML-templater
/includes/templates/admin	"Admin CP"-templater (kun HTML)
/includes/functions.py	"Utilities" / generelle funksjoner.
/includes/hp_init.spy	Spyce initialiseringsskript. Oppretter brukerobjekt, starter session, etc.
/sql	CREATE og INSERT SQL-uttrykk for MySQL DB.
/style	Stilark og -relaterte bilder (CSS).
/tinymce	TinyMCE 2.0.1 JavaScript-basert "RTF-editor" fra Moxiecode. Benyttet i Admin CP for enkel artikkelredigering. GNU/GPL lisensiert.
/xtm	Alle XML Topic Maps
/xtm/images.xml	XTM brukt i eksempler
/.htaccess	Apache HTTP Server direktiver

/2005.spy	Nettstedets "Utgravinger i 2005"
/about.spy	Nettstedets "Om hellerprosjektet"
/admin.spy	Nettstedets "Admin CP"
/asearch.spy	Nettstedets "Advanced search" (ikke ferdigstilt)
/contact.spy	Nettstedets "Kontakt oss"
/ErrorDocument403.html	"Error Document" (HTTP 403 Unauthorized)
/ErrorDocument404.html	"Error Document" (HTTP 404 Not Found)
/ErrorDocument500.html	"Error Document" (HTTP 500 Internal Server Error)
/hellere.spy	Nettstedets "Hellere på Vestlandet"
/images.spy	Nettstedets "Bilder fra hellere"
/index.spy	Nettstedets "Hjem"
/links.spy	Nettstedets "Eksterne ressurser"
/login.spy	Nettstedets "Logg inn"
/logout.spy	Nettstedets "Logg ut"
/map_vars.spy	XTM-parser klient for uthenting av variabel-verdier til /vestlandet.swf
/news.spy	Nettstedets "Nytt fra prosjektgruppen"
/printPage.py	Python-skript (Spyce-ekstensjon) for utskrift av WebPage-objekter.
/README.txt	Instruksjoner
/search.spy	Nettstedets "Søk versjon 0.1"
/showtopic.spy	Nettstedets "Emne" (via /showtopicmap.spy)
/showtopicmap.spy	Nettstedets "Emnekart" (via /showtopicmaps.spy)
/showtopicmaps.spy	Nettstedets "Emnekart"
/tema.spy	Nettstedets "Temaområder"
/usercp.spy	Brukerkontrollpanel (ikke implementert)
/vestlandet.swf	"Interaktivt" flash-kart for nettstedets "Hellere på Vestlandet". Viser informasjon fra emnekart. Se også /map_vars.spy

Referanser

Antoniou, G. and F. van Harmelen (2004). *A Semantic Web Primer*. The MIT Press.

Bergsvik, K. A., D. Elgesem, and A. K. Hufthammer (2004). Det vestnorske heller-prosjektet. Prosjektbeskrivelse, UiB.

Berners-Lee, T. (1991). World Wide Web Summary. `http://www.w3.org/Summary.html` (accessed Jan 30 2006).

Berners-Lee, T. (2004). Semantic Web "Layer Cake". `http://www.w3.org/2004/Talks/0412-RDF-functions/slide4-1.html` (accessed Jan 30 2006).

Berners-Lee, T., R. Fielding, and L. Masinter (1998). Uniform Resource Identifiers (URI): Generic Syntax. Request For Comments: 2396, Network Working Group.

Berners-Lee, T., J. Hendler, and O. Lassila (2001, May). The Semantic Web. *Scientific American 284*(5), 34–43.

Bush, V. (1945, Jul). As We May Think. *Atlantic Mothly 176*(1), 101–108.

Cagle, K. (2000). *XML Developer's Handbook*. Sybex.

Clark, D. D. (1995). The Design Philosophy of the DARPA Internet Protocols. *ACM SIGCOMM Computer Communication Review 25*(1), 102–111.

Copeland, J. (1993). *Artificial Intelligence. A Philosophical Introduction*. Blackwell Publishing.

Dictionary.com (2005). Dictionary.com. `http://dictionary.reference.com/` (accessed Jan 30 2006).

Ding, Y., D. Fensel, M. Klein, and B. Omelayenko (2002). The Semantic Web: yet another hip? *Data and Knowledge Engineering 41*, 205–227.

Ding, Y. and S. Foo (2002). Ontology research and development. part 1 - a review of ontology generation. *Journal of Information Science 28*, 123–136.

Dreyfus, H. L. (1979). *What Computers Still Can't Do. A Critique of Artificial Reason*. The MIT Press.

Elmasri, R. and S. Navathe (2004). *Fundamentals of Database Systems*. Addison Wesley.

Garshol, L. M. (2002). tolog. A topic map query language. `http://www.ontopia.net/topicmaps/materials/tolog.html` (accessed Jan 30 2006).

Garshol, L. M. (2003). Living with Topic Maps and RDF. `http://www.ontopia.net/topicmaps/materials/tmrdf.html` (accessed Jan 30 2006).

Garshol, L. M. (2004). Metadata? Thesauri? Taxonomies? Topic Maps! `http://www.ontopia.net/topicmaps/materials/tm-vs-thesauri.html` (accessed Jan 30 2006).

Garshol, L. M. (2005). TMQL. an introduction. `http://www.emnekart.no/2005/forum-04-19/tmql-intro.pdf` (accessed Jan 30 2006).

Google (2004). Google Technology. `http://www.google.com/technology/` (accessed Jan 30 2006).

Gruber, T. R. (1993). A Translation Approach to Portable Ontology Specifications. *Knowledge Engineering 5*(2), 199–220.

Guarino, N. (1998, Jun). Formal Ontology and Information Systems. In *Proceedings of FOIS'98, Trento, Italy*, pp. 3–15.

Haggarty, R. (2002). *Discrete Mathematics for Computing*. Addison Wesley.

Halmos, P. R. (1998). *Naive Set Theory* (1 ed.). Springer.

Hannemyr, G. (1999). *Begynnelsen på en historie om Internett*, pp. 11–27. Tano-Aschehoug.

Hoel, T. (2002). Bygges morgendagens læringsteknologi på gårdagens læringssyn? Master's thesis, IT University of Göteborg.

ISO (2002). ISO/IEC 13250 Topic Maps. International Standard, International Organization for Standardization.

ISO (2005). Topic Maps - Part 2: Data Model. Draft International Standard, International Organization for Standardization.

Jech, T. (2002). Basic Set Theory. `http://plato.stanford.edu/entries/set-theory/primer.html` (accessed Jan 30 2006).

Jing, Y. and W. B. Croft (1994). An Association Thesaurus for Information Retrieval. In *RIAO 94 Conference Proceedings*, pp. 146–160.

Jopp, C. (2004). Composing symphonies or singing karaoke? Norwegian perspectives on standardization. *Distances et savoirs 2*(4), 519–526.

Lacher, M. S. and S. Decker (2001). RDF, Topic Maps, and the Semantic Web. *Markup Languages: Theory & Practice 3*(3), 313–331.

McCarthy, J. (1987, Dec). Generality in Artificial Intelligence. *Communications of the ACM 30*(12), 1031–1035.

Miller, G. A. (1995, Nov). WordNet: A Lexical Database for English. *Communications of the ACM 38*(11), 39–41.

Moe, R. (2003). Bakgrunnsstoff for INFO121. `http://www.ifi.uib.no/undervisning/iv121/bakgrunn.pdf` (accessed Jan 30 2006).

Naughton, J. (1999). *A Brief History of the Future. The origins of the Internet*. Clays Ltd.

Newcomb, S. R. (2003). *A Perspective on the Quest for Global Knowledge Interchange*, Chapter 3, pp. 31–50. In Park and Hunting Park and Hunting (2003).

Nielsen, J. (2000). *Designing Web Usability*, Chapter 4, pp. 164–260. New Riders Publishing.

Noy, N. F. and D. L. McGuinness (2001, mar). Ontology Development 101: A Guide to Creating Your First Ontology. Technical report, Stanford Knowledge Systems Laboratory.

Ontopia (2003). The RTM RDF to topic maps mapping. Definition and introduction. `http://www.ontopia.net/topicmaps/materials/rdf2tm.html` (accessed Jan 30 2006).

Park, J. and S. Hunting (Eds.) (2003). *XML Topic Maps. Creating and Using Topic Maps for the Web*. Addison-Wesley.

Passin, T. B. (2004). *Explorer's Guide to the Semantic Web*. Manning.

Pepper, S. (2000a). The TAO of Topic Maps. `http://www.ontopia.net/topicmaps/materials/tao.html` (accessed Jan 30 2006).

Pepper, S. (2000b). Ten Theses on Topic Maps and RDF. `http://www.ontopia.net/topicmaps/materials/rdf.html` (accessed Jan 30 2006).

Pepper, S. and L. M. Garshol (2002). The XML Papers. Lessons on Applying Topic Maps. `http://www.ontopia.net/topicmaps/materials/xmlconf.html` (accessed Jan 30 2006).

Pepper, S. and S. Schwab (2002). Curing the Web's Identity Crisis. Subject Indicators for RDF. `http://www.idealliance.org/papers/dx_xmle03/papers/02-03-04/02-03-04.pdf` (accessed Jan 30 2006).

Pepper. S and Graham M. (Eds.) (2003, mar). XML Topic Maps (XTM) 1.0 Specification. TopicMaps.Org Specification, TopicMaps.Org.

Pepper, S. (Ed.) (2003, jun). Published Subjects: Introduction and Basic Requirements. Available via `http://www.oasis.org/` (accessed Jan 30 2006).

Pitti, D. V. (2004). Standard Generalized Markup Language and the Transformation of Cataloging. `http://xml.coverpages.org/berknasg.html` (accessed Jan 30 2006).

Powers, S. (2003). *Practical RDF*. O'Reilly.

Rittershofer, A. (2005). *Supporting Self-regulated E-Learning with Visual Topic-Map-Navigation*, pp. 355–363. Springer-Verlag GmbH.

Shiflett, C. (2003). *HTTP Developer's Handbook*. Sams Publishing.

Slomin, D. and R. Tengi (2005). WordNet 2.1 Browser. `http://wordnet.princeton.edu/` (accessed Jan 30 2006).

Sowa, J. F. (2000). *Knowledge Representation. Logical, Philosophical, and Computational Foundations*. Brooks/Cole.

Uschold, M. (2001). Where is the Semantics in the Semantic Web? In *Workshop on Ontologies in Agent Systems (OAS) at the 5th International Conference on Autonomous Agents*.

W3C (1998). Resource Description Framework (RDF). W3C Recommendation, World Wide Web Consortium.

W3C (2004a). Extensible Markup Language (XML) 1.0 (third edition). W3C Recommendation, World Wide Web Consortium.

W3C (2004b). OWL Web Ontology Language Overview. W3C Recommendation, World Wide Web Consortium.

W3C (2004c). RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation, World Wide Web Consortium.

W3C (2004d). XML Schema Part 0: Primer Second Edtion. W3C Recommendation, World Wide Web Consortium.

W3C (2005). W3C Semantic Web. `http://www.w3.org/2001/sw/` (accessed Jan 30 2006).

Walsh, N. (2005). *DocBook: The Definitive Guide 0.0.12 for DocBook V5.0b1*. O'Reilly & Associates, Inc.

Winograd, T. (1970). SHRDLU. `http://hci.stanford.edu/~winograd/shrdlu/` (accessed Jan 30 2006).

Wolff, K. E. (1993). A First Course in Formal Concept Analysis. *Advances in Statistical Software 4*, 429–438.

Wooldridge, M. (2002). *An Introduction to MultiAgent Systems*. John Wiley & Sons Ltd.