

UNIVERSITETET I OSLO
Institutt for informatikk

Flerspråklighet i IT

Mads Gøran Sørhaug

Mai 2005



Forord

Oppgaven er en mastergradsoppgave ved Institutt for Informatikk, Universitetet i Oslo.

Hovedfagsveilederen min ved Institutt for informatikk har vært Odd Aurmo. Ekstern veileder har vært Sverre Bauck fra Brønnøysundregistrene. Jeg ønsker å takke dem begge for uvurderlig hjelp, uten dere hadde dette ikke vært mulig.

Jeg må også takke til Brønnøysundregistrene generelt for at jeg fikk jobbe med denne oppgaven og Risten Turi hos Sametingets språkavdeling for hjelp med den samiske språkdelen.

Innholdsfortegnelse

Forord	1
Innholdsfortegnelse	3
Figurer	4
Tabeller	5
1 Innledning	7
1.1 Kapitteloversikt.....	8
2 Bakgrunn	9
2.1 Kultur- og kyrkjedepartementet om mållova, nettsider og elektroniske meldingar .	9
2.2 Altinn	11
2.2.1 Inndeling	11
2.2.2 Skjema.....	11
2.3 Språk	12
2.3.1 Samisk.....	13
2.3.2 Andre språk.....	14
2.4 Brukerkrav	15
2.5 Nytte og behov	16
3 Forutsetninger for oversetting	19
3.1 Tegn	19
3.2 Fra tegn til informasjon.....	22
3.3 Sortering.....	22
3.4 Kostnader ved sortering	24
3.5 Verktøy	26
3.5.1 Nettlesere	26
3.5.2 Domenenavn	26
3.6 Bruk av Unicode	28
4 Oversetting	31
4.1 Automatisk oversetting av fritekst	31
4.1.1 Nyno.....	32
4.2 Oversetting for hånd	32
4.3 Oversetting av kodbar informasjon - ordbokmetoden	33
4.4 Oversetting av kodbar informasjon – feltløsning.....	34
4.5 Bruksområder for skjematisk oversettelse	37
5 Praktisk eksempel – Altinn	39
5.1 Kravspesifikasjon.....	39
5.1.1 Språk	39
5.1.2 Brukerkrav	40
5.1.3 Tegnssett.....	40
5.1.4 Sortering.....	41
5.1.5 Oversetting.....	42
5.1.6 Andre vurderinger	43
5.2 Feltløsning for Altinn.....	44
5.2.1 Språktabell	44
5.2.2 Sortering.....	45
5.2.3 Oversetting.....	51

5.3 Beskrivelse av programvare.....	56
6 Kost/Nytte Analyse.....	57
7 Andre systemer og løsninger.....	61
7.1 Acharya.....	61
7.1.1 Font basert data innsetting.....	62
7.1.2 Translitterasjon basert på data innsetting.....	62
7.1.3 Manuell skrivemaskin innsetting.....	62
7.1.4 Spesielle programmer for å lage websider.....	63
7.1.5 Phonetisk kartlegging av vokaler og konsonanter.....	63
7.2 Flerspråklige løsninger.....	63
7.2.1 Norge.no.....	63
7.2.2 Canada.ca.....	67
7.2.3 Norge.no vs Canada.ca.....	71
7.2.4 Min Side.....	72
7.2.5 Skolelinux.....	73
7.2.6 Andre eksempler.....	74
7.3 IGLO prosjektet.....	74
8 Oppsummering og konklusjon.....	77
8.1 Generelle tilrådninger.....	77
8.2 Tilrådning for Altinn.....	79
Vedlegg A: Kildekode.....	81
Sortering.....	81
Feltløsning.....	84
Vedlegg B: Referanser.....	91

Figurer

Figur 1: Statsorgan med mer enn 25 % nynorsk – i prosent av respondentene.....	10
Figur 2: Gjennomsnittlig prosent nynorsk.....	10
Figur 3: Insert-sort.....	25
Figur 4: http://www.språkrådet.no i IE 6.0 uten tilleggspakke.....	27
Figur 5: Ord kan ha flere betydninger. Her noen norske oversettelinger av det engelske ordet frame.....	34
Figur 6: Enkel e-post, her på bokmål.....	35
Figur 7: Forside av førerkort, EØS-modell. På høyre side skimter man så vidt førerkort skrevet på de forskjellige språkene. I midten, til høyre for bildet, ser man de feltene med personopplysninger.....	38
Figur 8: Logg inn hos Altinn. Bildet gjort skarpere for at det skulle bli mer lesbart i dokumentet.....	51
Figur 9: Header i Altinn med språkvalg.....	53
Figur 10: Skjerm bilde av eksempelprogrammet. Viser "Logg inn" siden på bokmål.....	54
Figur 11: Skjerm bilde av eksempelprogrammet. Viser "Logg inn" siden på samisk.....	55
Figur 12: Norge.no.....	65
Figur 13: Noreg.no.....	66

Figur 14: Norway.no.....	67
Figur 15: Forsiden til Canda.ca eller http://canada.gc.ca	68
Figur 16: Den engelske forsiden til Canada.ca http://canada.gc.ca/main_e.htm	69
Figur 17: Den franske forsiden til Canada.ca. http://canada.gc.ca/main_f.html	70

Tabeller

Tabell 1: ISO kodeliste over språk.....	16
Tabell 2: ASCII -I.....	20
Tabell 3: Små og store bokstaver i det nordsamiske alfabet i sortert rekkefølge. (Bokstaver som finnes i fremmedord er i parentes.)(Trosterud 2002:Samiske bokstaver)(Unicode)	23
Tabell 4: Ved å tilegne ord en verdi så kan man skille mellom betydningen av samme ord. Her skilles betydningen av det engelske ordet frame ved å gi den to verdier, som tilsvarende ordene ramme og stativ på norsk.	34
Tabell 5: Mulig skjema for flerspråklig løsning av enkel e-post. I tillegg til feltene tatt med i tabellen, er også feltene man skriver i viktige for løsningen. Informasjonen der må kunne tydes av mottaker. Det er derfor mer egnet med numeriske verdier og valgli ster enn fritext her. Dermed blir e-posten et dårlig eksempel.	35
Tabell 6: Hvert felt består av 10 tegn. Størrelsen er avrundet for lettere lesning. Filen er en tekstfil i Windows 2000 NTFS filsystem og målingene er gjort på en AMD64 3200, 1024MB RAM.	37
Tabell 7: Sorteringstabell for nordiske tegn, slik tegn vil sorteres på bokmål, nynorsk og samisk.	44
Tabell 8: Felttabell av Altinn sin logg inn side. Studerer man siden, ser man at det blant annet er noen linker der. Dette betyr at en del av tekstene i tabellen skulle bestått av html-/javascriptkode i stedet for bare ren tekst. Dette er ikke tatt med av hensyn til lesbarheten av tabellen. Feltene 9 og 10 er feltene der fnr og passord skrives inn og står derfor tomme i utgangspunktet. Feltene merket med * vises avhengig av hva som er valgt i feltene 6 og 7.....	52
Tabell 9: Kolonnen for samisk. Oversatt av Risten Turi hos Samisk Språkråd. Turi gjorde oppmerksom på at oversettelsen ikke var korrekturlest, og at noen termer som ble lagt i forbindelse med oversettelsen ikke var kvalitetssikret.	53
Tabell 10: Etatsdialog i Altinn – kost/nytte.	57

1 Innledning

Helt siden Ivar Aasen introduserte landsmål (i dag: nynorsk) har diskusjonen om flerspråklighet vært aktuell her i landet. De siste årene har utbredelsen av IT og Internett preget verden. Mange offentlige og private tjenester er nå blitt tilgjengelige gjennom dette mediet, og med dette har diskusjonen om flerspråklighet nådd et nytt marked.

For å kunne kommunisere, må kommunikantene forstå hverandre. I mange tilfeller vil de kunne forstå hverandre skriftlig, men ikke muntlig, som for eksempel for brukere av mandarin kinesisk. For brukere av store språk er antallet de kan kommunisere med viktig. For mindre språk er nyanserikdommen overordnet antallet man deler språk med. I det siste har oppfatningen at det å kunne gjøre seg forstått er viktigere enn at alle deler samme språk. Tankegangen har flyttet seg fra et stort felles språk, til å øke forståelsen mellom folk som snakker ulikt. Forskjellen mellom dansk, norsk og svensk er sannsynligvis mindre enn mellom tysk fra Sveits til Hamburg. Likevel regnes det som tre forskjellige språk.

Oppgaven omhandler utfordringene ved å lage gode flerspråklige systemer. Hvordan utfordringene kan løses i enkelte tilfeller, med Altinn som et eksempel på dette. Oppgaven bruker stort sett gjenge begreper, og kan oppfattes som en liten del av globaliseringsdiskusjonen.

Tegn og tegnssett er den første utfordringen som blir drøftet. Unicode og andre tegnssett har alle sine fordeler og ulemper, og må vurderes i de enkelte tilfellene. Sortering av særtegn som Æ, Ø og Å ble så drøftet med de vansker dette medfører. Spesielle tilfeller som for eksempel at dobbel a på norsk kan sorteres som Å medfører noen ekstra vanskeligheter.

Oversetting er et stort tema i oppgaven. Det skilles mellom oversetting av fritekst og oversetting av kodbar informasjon av den enkle grunn at automatisk oversetting av fritekst er en tilnærmet umulig oppgave for et dataprogram. Feltløsningen fremtrer som en egnet måte for oversetting i en rekke systemer. Løsningen benytter kun maskinvare og software som er tilgjengelig i dag. Løsningen tilbyr en flerspråklig løsning, med brukervennlig språkskifte, som kan utvides med et uendelig antall språk. Tankegangen bak er svært enkel, og implementasjonen er også enkel dersom forutsetningene ligger til rette.

Eksempelprogrammet viser logg-inn siden hos Altinn implementert i Java applikasjon, og omgjort til å inneholde en feltløsning. Det er også implementert et eksempel på hvordan sortering kan gjøres med samiske og norske tegn.

Til slutt blir det gitt noen anbefalinger, generelle og til Altinn. Disse omhandler hva man bør ta hensyn til når man skal lage et flerspråklig system.

1.1 Kapitteloversikt

Kapittel 1 inneholder innledningen, samt beskrivelsen av Altinn, språkene som er ønsket i Altinn sin flerspråklige løsning, brukerkrav til flerspråklig løsning og hvilken nytte og behov man har av en slik løsning.

Kapittel 2 beskriver forutsetningene for oversetting. Her blir det drøftet forutsetningene som må være på plass før oversettingen kan gjennomføres. Tegnsett og sortering av disse er i fokus.

Kapittel 3 inneholder metoder for oversetting. Her drøftes automatisk oversetting av fritext, kodbar informasjon, ordbokmetoden, og sist men ikke minst feltløsningen.

Kapittel 4 inneholder et praktisk eksempel på hvordan en flerspråklig løsning kan implementeres i Altinn. Feltløsningen blir her brukt på ”logg inn” siden til Altinn.

Kapittel 5 inneholder en kost/nytte analyse av et flerspråklig system

Kapittel 6 inneholder en beskrivelse av andre systemer og løsninger. Hva andre har gjort, og en drøftning av dette mot feltløsningen.

Kapittel 7 inneholder en oppsummering og konklusjon, generelle tilrådninger og tilrådninger for Altinn.

2 Bakgrunn

NOU nummer 13 (1997-98) *Målbruk i offentlig teneste* omhandler norsk språk og informasjons- og kommunikasjonsteknologi (IKT). ”Det trengs eit offensivt handlingsprogram der både juridiske, administrative, språkteknologiske og økonomiske verkemiddel blir tekne i bruk. Målet er å kunne gi både nynorsk- og bokmålsbrukarane gode reiskapar til støtte for språkbehandling og å tilby relevante informasjonstenester o a på begge målformer” (s. 29). Dette førte vidare til *Handlingsplan for norsk språk og IKT* der hovedmålene er:

1. I Norge skal IKT-produkt og -tjenester ha norskspråklig brukergrensesnitt som bygger på kunnskap om norsk tale og de to målformene
2. Alle med norsk som morsmål skal stilles likt i tilgang til og valg av norskspråklige IKT-produkt og -tjenester og skal gis like muligheter til å skaffe seg kompetanse i bruken av dem
3. Mengden norskspråklig programvare må økes, særlig programvare på nynorsk
4. Opplæringslovens bestemmelser om språklige parallellutgaver må oppfylles i praksis for elektroniske læremiddel
5. Språkteknologisk forskning og utvikling i Norge skal skje på norskspråklig grunn (bokmål, nynorsk og dialektene) og må tilføres tilstrekkelige økonomiske midler
6. En standardisert norsk IKT-terminologi må utvikles og stilles til rådighet
7. Norge skal delta aktivt i det nordiske og øvrige europeiske samarbeidet innenfor språkteknologi
8. Norge skal bidra til å styrke erfaringsutvekslingen mellom språkorganisasjonene i Norden og ellers i Europa om informasjons- og kommunikasjonsteknologiens innvirkning på nasjonalspråkene og om mottiltak for å holde språkene i hevd også på dette området

(Norsk Språkråd juni 2001)

2.1 Kultur- og kyrkjedepartementet om mållova, nettsider og elektroniske meldingar

Fleire statsorgan har eigne sider på Internett, og det tilfanget som blir lagt ut der, blir også omfatta av reglane i mållova. Det er viktig at ikkje berre ein del av det vekslande tilfanget er på nynorsk, men at også ein del av dei faste tekstane og menyane blir laga på nynorsk, slik at begge dei offisielle målformene er klart synlege på nettsidene.

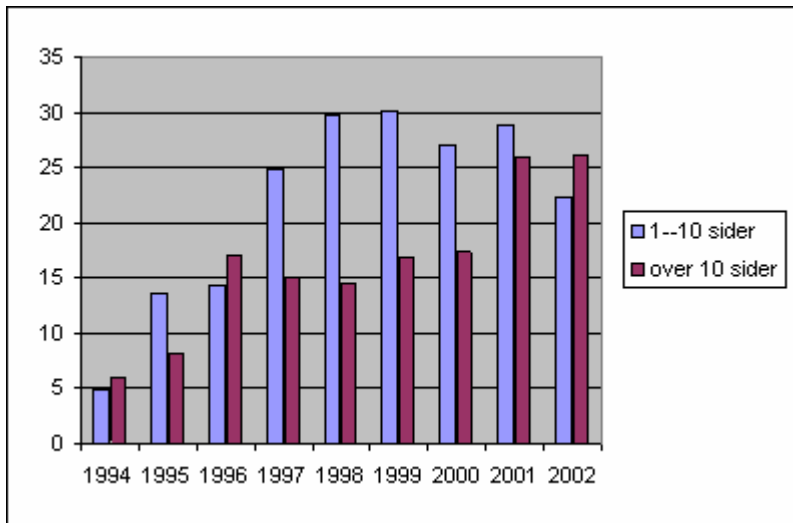
NOU nummer 9 (2001–2002)

Dersom eit statstilknytt organ korresponderer med einskildpersonar, andre private rettssubjekt eller med kommunar og fylkeskommunar på grunnlag av eit elektronisk system som produserer standardiserte meldingar eller svar, skal dei standardiserte tekstane finnast både i ein bokmåls- og ein nynorskversjon, og systemet skal kunna registrera kva målform den einskilde korrespondansepartnar brukar eller ynskjer, og

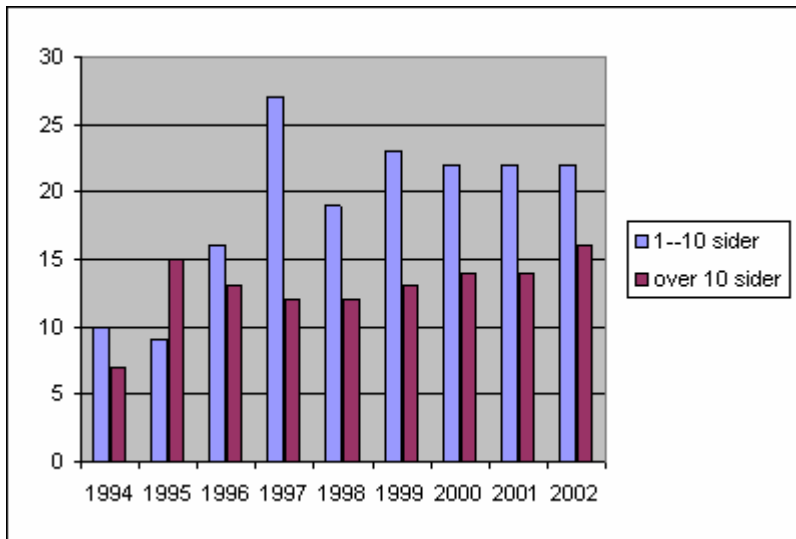
vera innretta slik at dei automatisk får standardmeldingane i den målforma dei er registrerte med. Dette kan til dømes dreia seg om kunde- eller medlemsregister, men det kan vera kva som helst anna slags register eller system.

Dersom eit statstilknytt organ driv aktivitetane sine over heile landet eller i prinsippet vender seg til heile landet, skal allment informasjonstilfang som vert lagt ut på Internett, avspegla at landet har to offisielle målformer, anten ved at tekstane finst både i bokmåls- og nynorskversjon, slik at ein velja målform ved å klikka på bokmål eller nynorsk, eller ved at minst fjerdeparten av tekstane er på nynorsk, eventuelt på bokmål.

NOU nummer 48 (2002–2003)



Figur 1: Statsorgan med mer enn 25 % nynorsk – i prosent av respondentene



Figur 2: Gjennomsnittlig prosent nynorsk

(Norsk språkråd)

Figurene viser at selv om prosenten av statsorganene med mer enn 25 prosent nynorsk ser ut til å øke, så øker ikke andelen nynorsk på landsbasis noe merkbart etter 1996.

2.2 Altinn

Altinn er en portal - et felles nettsted - for dialog med det offentlige Norge.

Altinn ble etablert for å være en innrapporteringskanal til Skatteetaten, Statistisk Sentralbyrå, og Brønnøysundregistrene og var resultatet av et samarbeid mellom flere offentlige etater som innhenter data fra bedrifter og privatpersoner. Det var disse som med sterk støtte fra NHD som tok initiativ til Altinn. Senere er også Statens Lånekasse, Konkurransetilsynet, Norges Bank, Fiskeri- og kystdepartementet, Statens Innkrevingsentral, Økokrim, Produktregisteret og Kredittilsynet kommet med.

Grunntanken bak Altinn er at offentlige skjemaetater samarbeider om å lage en felles kanal for dialog med det offentlige i stedet for å lage hver sin løsning. "En kanal - ett sted å forbedre".

Målet er å komme frem til en situasjon der alle skjemaer er bygd over samme lest, slik at brukerne slipper å måtte lære seg/forholde seg til flere ulike brukergrensesnitt. Etatene skal kunne samarbeide for å få bedre kvalitet og dermed mer effektiv kommunikasjon med landets næringsliv og innbyggere.

2.2.1 Inndeling

Altinn er inndelt i fire hoveddeler, en portal, en formidlingsdel, dialog og en sikkerhetsdel. Portalen brukes i hovedsak til å navigere mellom skjema. Den er bygd på asp teknologi, noe som gjør den utvidbar med tanke på nye funksjoner som flerspråklighet. Sikkerhetsdelen håndterer blant annet autentisering, autorisasjon, signering, og senere kryptering.

2.2.2 Skjema

Gjennom skjemaene skjer nå innrapporteringen/dialogen. Disse har flere leverandører, men alle skjema som benyttes av Altinn baserer seg på formatet til Oppgaveregisteret. Skjemaene defineres etter XML Schema (XSD) standarden. Bedriftenes administrative systemer benytter seg av disse XSD definisjonene til å bygge opp og validere meldingene. Dataene vil også bli validert av Altinn basert på de samme definisjonene. Webservices er også i ferd med å bli implementert.
(Altinn)

Skjema basert på XML teknologi gjør at disse er lette å utvide, det er mulig å lage ganske forskjellige skjema uten at systemet får problemer med å vise dem på ønsket vis. Ved å bruke webservices og XML basert teknologi sørger Altinn for at strukturen

- Er sikker for avsender og mottaker
- Benytter aksepterte Internett standarder
- Unngår bruk av klient/modul hos fagsystem, som gir en sømløs integrasjon.
- Er uavhengig i forhold til fagsystemets tekniske plattform (Fagsystem)



Figur 3: Overordnet arkitektur

2.3 Språk

Norsk tilhører den germanske språkfamilie. Det har blitt skrevet med det latinske alfabet siden 1000-tallet. Det gammelnorske skriftspråket gikk av bruk under svartedauden i 1349 og ble da erstattet av dansk. Etter 1814 vokste det frem et sterkt ønske om å fornorske skriftspråket. Wergeland, Asbjørnsen og Moe, Bjørnson og Ibsen bidro vesentlig til det. Fornorskningen tok i annen halvdel av 1800-tallet to veier: Knud Knudsen ønsket å fornorske det dansk-norske språket gradvis på alle felt. Ivar Aasen satte opp en ny språkreform på grunnlag av dialektene. Disse to språkformene ble kalt henholdsvis riksmål (fra 1929 bokmål) og landsmål (fra 1929 nynorsk). De ble sidestilt som offisielle språkformer i 1885.

Det ble vedtatt en rekke rettskrivningsendringer i første halvdel av 1900-tallet. Disse tok sikte på en tilnærming mellom de to målformene. ”Samnorsk-linjen”, som dette ble kalt, møtte sterk motstand, og ble i 1960-årene avløst av en språkfredspolitik som har nedprioritert en slik styring av språkutviklingen. I 1990-årene hadde rundt 16 prosent av elevene i grunnskolen nynorsk som hovedmål.

Nynorsk har gjennomgått store endringer og dagens nynorsk er svært forskjellig fra den fra før 1970. De to språkene er i dag så like at de av mange anses for forskjellige presentasjonsformer av samme språk. Forskjellen i grammatikk og syntaks er i praksis visket ut.

Den dag i dag er det fremdeles blest om nynorsk. I desember 2004 fremmet Høyre et ønske om å gjøre nynorsk til valgfag i grunnskolen. Venstre og KrF var sterkt imot dette. I dag er det ni videregående skoler i Oslo som har forsøk med valgfritt sidemål.

”Tilbakemeldingene er veldig positive. Elever og lærere føler de får mer tid og ro til fordypning i hovedmålet, og sidemålet blir mer lystbetont og morsommere. Motstanderne sier at sidemålet vil forsvinne. Tvert om er interessen for å lære større”, sier Marie Eriksen i Høyre.

”Vi er uenige om dette. KrF ønsker å styrke nynorsken. Å gjøre det til et valgfag vil svekke nynorsk. Det vil være et farvel til sidemålet”, sier KrFs nestleder Knut Arild Hareide til avisen VG.

2.3.1 Samisk

Samisk er egentlig en gruppe uraliske språk eller dialekter som tales fra Kolahalvøya til Engerdal i Hedmark. Det kan deles inn i østsamisk, sentralsamisk og sørsamisk. Disse grupperingene kan igjen deles inn i mindre undergrupper. Ca 90 % av de samisktalende snakker nordsamisk som er en undergruppe av sentralsamisk. Samisk Språknemnd (Sámi Giellalávdegoddi) vedtok i 1978 en felles nordsamisk rettskriving for Norge, Sverige og Finland. Samisk ble i 1992 offentlig språk i Norge ved siden av norsk. (Caplex)

Mange samer synes at det er enklere å lese norsk enn samisk. Spesielt når det gjelder offentlige dokumenter, eller andre ting med ”moderne” språk. Til det mener Risten Turi ved Samisk Språkråd at det er det tre hovedgrunner:

”En stor andel av voksne samer (ca. 50 %) er så å si analfabeter når det gjelder sitt eget språk. De fikk ikke opplæring i samisk, og har dermed ikke fått muligheten til å lære, lese og skrive samisk.

Når man skal oversette tilbud slik som Altinn til samisk så krever det en god del ny terminologi som må lages. På sidene til Altinn kan det nevnes ord som passord, identitet, funksjonalitet, PIN-kode og mobiltelefon. I Norge er norsk hovedspråket, det vil si at alt som er nytt blir lansert på norsk først. Det gjelder ikke bare offentlige tjenester men også

nye varer og tjenester. Noen eksempler er bærbar PC, mobiltelefon, iPod. Det skjer aldri at man venter med å lansere nye ting til man har et samisk navn på denne tingen. Samiske språkarbeidere vil dermed alltid være på etterskudd. Dette fører igjen til at samene må ta i bruk det norske ordet først, rett og slett fordi man mangler samiske ord. Vanlige mennesker lager veldig sjelden ny terminologi selv, da må man i så tilfelle være spesielt interessert. (Dette skjer ikke i norsk heller). Så når samiske oversettere og språkarbeidere lager samiske ord for ting som bærbar PC, mobiltelefon og iPod, så vil de virke veldig kunstige og fremmede fordi samene er vant til å bruke de norske ordene. Den samme mekanismen finner man i norsk.

Samisk er veldig lite synlig i det norske samfunnet. På nordsamisk finnes ingen dagsaviser og vi har bare 15. min nyhetssending på det lisensfinansierte NRK hver dag. For sørsamisk og lulesamisk er tilbudet enda verre; der finnes det ingen aviser og ingen nyhetssendinger overhodet. Det finnes ikke daglige barne-TV eller ungdomssendinger på TV. All denne mangelen gjør at samisk for mange blir et hjemmespråk, og når man bare bruker et språk hjemme, så blir det ikke så veldig mye bruk for ord som offentlige tjenester, tjenestemann, forskrifter, passord osv. Når man da lager ord for disse ordene så virker de fremmede fordi man ikke er kjent med ordene. Eks: vet du hva disse ordene heter på norsk: hat trick, hacker, keyboard, pop-up meny, widescreen, og ville du begynt å bruke de norske ordene om du fikk vite hva de heter på norsk?

Når det gjelder offentlige tjenester på samisk så er det viktig å huske på at det bare er de siste årene at man har begynt å tilby dette. Det betyr at mange samer må venne seg til at de kan bruke samisk. Og de må også få informasjon om at tjenestene nå er tilgjengelig på samisk. Det har liten hensikt å tilby tjenester på samisk om man ikke jobber med å gjøre denne tjenesten kjent for den samiske befolkningen. Og jeg har den oppfatningen at bruken av samiske tjenester vil ta seg opp med årene, fordi samene blir kjent med at de finnes og fordi man med tiden blir kjent med den nye samiske terminologien. I dette arbeidet er det viktig å ha en offensiv holdning for ellers vil ikke det samiske språket utvikle seg til å være et språk man kan bruke i det moderne samfunnet.”

2.3.2 Andre språk

Kvenene er en etnisk gruppe med tilhørighet i Nord-Norge. Befolkningsgruppen har siden 1998 vært anerkjent som en nasjonal minoritet i Norge. Kvensk språk brukes av 2000-8000 personer, avhengig av hvilke kriterier som legges til grunn for beregningen. Kvensk anerkjennes som eget språk med fortsatt vern på nivå II i henhold til den europeiske minoritetsspråkpakten.

I Kirkenes har det etter hvert etablert seg en stor gruppe med russisk som morsmål. Innvandringen fra Russland fortsetter, mens fraflyttingen av folk med norsk som hovedmål fremdeles er sterk. Russlands økonomi bedrer seg, og avgiftssystemet nord i landet er svært gunstig. Mange rike russere kommer til Norge og Kirkenes for å handle. Dette ligner på svenskehandelen som foregikk mens det var restriksjoner i Norge.

Allerede har det kommet vegskilt på russisk, men behovet for større støtte av de statlige tjenestene på russisk kan melde seg raskere enn vi aner.

Engelsk er av mange ansett som et Lingua Franca, et språk som de fleste i en region skjønner. Mange vil derfor ha med engelsk i sine flerspråklige løsninger. I store deler av Europa har derimot fransk eller spansk rollen som Lingua Franca.

”Ved inngangen til 2004 var innvandrarbefolkninga i Noreg på om lag 349 000 personer. Dette tilsvarte 7,6 prosent av den totale folkemengda.

Innvandrarbefolkninga auka med 16 100 personar, eller 4,9 prosent, i 2003. Til samanlikning var auken for heile den norske befolkninga 25 200”

(Statistisk sentralbyrå).

Med slike tall er det tydelig at det finnes mange i dette landet som verken snakker samisk eller norsk. Innvandrerspråkene må tas hensyn til dersom man skal vurdere fremtidige utvidelser. Det vil selvfølgelig være en særdeles vanskelig, for ikke å si umulig oppgave å lage systemer som støtter alle verdens språk. Imidlertid er enkelte systemer som vil tjene på å støtte flest mulig språk. Lovverket er et eksempel på dette. Alle som kommer til Norge burde ha muligheten til å skjønne og følge de norske lovene.

2.4 Brukerkrav

Et av de viktigste argumentene for å lage et språkuavhengig system er at brukeren skal kunne bruke systemet uavhengig av hvilket språk han/hun snakker. Dette virker kanskje logisk, men for at brukeren skal oppfatte systemet som virkelig språkuavhengig, må språkbyttet skje så lett og så tidlig som mulig. Dersom brukeren må gjennom mange steg av prosessen før byttet skjer, er det større sannsynlighet for at han/hun misforstår noe eller aldri får gjennomført noe bytte. Derfor bør språkvalget med påfølgende språkbytte være noe av det aller første som skjer. Aller helst bør dette skje allerede før innlogging, men dersom dette ikke er mulig, må systemet være på riktig språk når innloggingen er gjennomført.

Første gang man besøker en side kan det være vanskelig å vite hvilket språk man ønsker. Derfor må som regel brukeren velge språket selv. For å gjøre et slikt bytte enklere bør man indikere språkvalgene med internasjonale koder, ISO har utgitt slike kodelister. Man ser ofte at flagg brukes, men det er få land som er enspråklige, og flagget gir ingen entydig henvisning til et språk. Selv om samene har eget flagg, skiller ikke det norske flagget mellom bokmål og nynorsk. I stedet for å skrive for eksempel ”engelsk”, kan det også være en fordel at det står ”english”.

Språk	ISO 639-2	ISO 639-1
Norsk	nor	no
Norsk (Bokmål)	nob	nb
Norsk (Nynorsk)	nno	nn
Engelsk	eng	en
Samisk (Nordsamisk)	sme	se

Tabell 1: ISO kodeliste over språk

En annen ting som er viktig å passe på er å opprettholde konsistens på tvers av språkene. Det vil si at sidene må være likt oppbygd på de forskjellige språkene. Siden det er svært vanskelig å få en side helt identisk på to språk som er noe forskjellig, som for eksempel bokmål og arabisk, er man nødt til å lette litt på kravene. Et mål på konsistensen kan være i hvilken grad brukeren oppfatter sidene som identiske. Dersom bokmålsbrukeren finner det han leter etter på det stedet han forventer i nynorskutgaven, virker siden konsistente for denne brukeren.

For å utdype behovet for dette med et eksempel, er det svært vanlig i slike systemer at det blir ferdigstilt med ett, eller noen få språk til å begynne med, la oss si nynorsk. En bokmålsbruker som har ventet lenge på systemet vil da kanskje benytte den nynorske utgaven med en gang den kommer. Når bokmålsutgaven så kommer, og ikke likner på nynorskutgaven, vil nok brukeren bli forvirret og mest sannsynlig ikke se noen forbedring. Bokmålsutvidelsen vil være bortkastet på denne brukeren. Et annet tilfelle kan være dersom en nynorskbruker vil henvise til noe til bokmålsbrukere. Dersom det er dårlig konsistens på sidene kan det i beste fall bli svært vanskelig å finne frem til dette punktet.

2.5 Nytte og behov

Alt man vil gjøre koster penger. Det er liten tvil om at mange gjerne ville sett, blant annet Altinn, på både nynorsk, samisk og engelsk, men det finnes grunner til at dette ikke har blitt prioritert. Svært få bedrifter prioriterer å få flerspråklighet på plass, dersom de ikke ser en større økonomisk gevinst med dette. Slik er det til en viss grad i det offentlige også. Finnes det så noen økonomiske fortjenester med å ha en god flerspråklig innrapporteringstjeneste i Norge, og er den i så fall så stor at man kan forsvare å bruke mange penger på å få på plass denne løsningen?

En stor del av Skatteetatens oppgaver i forbindelse med innlevering av selvangivelse, og annen innrapportering, er å oppklare missforståelser som utfyller har gjort. I følge Lise Halvorsen i Skatteetaten ser de ikke selv på dette som feil i og for seg, men som en del av jobben de er satt der for å gjøre. Derfor finnes det heller ingen tall over feil som er gjort. Likevel vil en flerspråklig løsning bidra til at feilprosenten går noe ned.

Både selvangivelsen og andre skjema fra Skatteetaten finnes i Altinn på nynorsk, men siden resten av Altinn ikke har nynorsk løsning, og siden det ikke finnes noen samisk eller engelsk løsning foretrekker kanskje mange å benytte papirløsningene over den elektroniske. Dette gjør at etaten må bruke en del ressurser på å føre inn det som blir rapportert skriftlig inn i datasystemene. Dersom man kan automatisere denne rapporteringen, også på flere språk, kan man spare mye på dette. Statistisk Sentralbyrå kan kutte ned på antall undersøkelser som blir sendt ut skriftlig dersom man kan benytte en portal i stedet. Svarene deres blir mer presise slik at man også her slipper unna mye feil. Man slipper å forkaste resultater, eller be utfyller gjøre arbeid flere ganger. Når dette skjer via post, koster det også penger. Lånkassen og andre banker samt kundene deres kan tjene på løsninger ved for eksempel søknad på lån skjer mye raskere enn det ville gjort ellers. Med flerspråklighet i dette tilfelle, ville en ren samisktalende slippe å ha med tolk for å søke på lån, og ville i tillegg kunne fått svaret på dagen.

En bruker som kan benytte sitt eget språk, vil som regel foretrekke dette fremfor et språk han ikke kan uttrykke seg så godt i. Dette vil igjen kanskje føre til at han er mindre motvillig til å gjennomføre en slik undersøkelse, eller vil bruke banken som tilbyr løsninger på hans språk. Fokus bank var den eneste landsdekkende banken som kunne tilby nynorsk løsning. Frem til november 2001 kunne banken tilby kundene et valg mellom bokmål og nynorsk. Tospråkligheten var historisk betinget og knyttet til at Vestlandsbanken, som var ren nynorskbank, var en av fire banker som i 1987 dannet Fokus Bank. Datasystemet som lå til grunn var felles for en rekke banker som kjørte på Norddata/Vestdata. Norddata/Vestdata ble senere til Novit, Telenor Novit, EDB Fellesdata, som nå ligger under EDB Business Partner ASA. Bankene eide den felles datasentralen og flere av bankene, blant annet lokale sparebanker på Vestlandet brukte nynorsk.

På 90-tallet var det omkring 15000 kunder i Fokus som fikk informasjon på nynorsk (av ca 200 000 kunder totalt). Da Fokus Bank høsten 2001 gikk over til Danske Banks eget datasystem, ble det utredet å videreføre nynorsk i tillegg til bokmål. Dette ville kreve omfattende oversettelser i grunnsystemene, og de fant det ikke økonomisk forsvarlig å insistere på dette overfor sine danske eiere. Fokus bruker fortsatt nynorsk i individuelle brev og en del markedsmateriell i nynorskområdene på Vestlandet.

De beste argumentene for flerspråklighet er likevel andre enn økonomiske. Offentlige tjenester på samisk sparer man ikke mange kroner på. 138 personer ba i år om å få selvangivelsen på samisk. Dersom disse hadde gjort samme arbeid gjennom Altinn ville man spart $16 \text{ NOK} * 138 = 2208 \text{ NOK}$ (Se kap. 5 – Kost/nytte analyse). Ikke en voldsom besparing. At folk benytter en god løsning er ofte den beste reklamen man kan få. Som Risten Turi nevnte tidligere er det viktig at man opplyser om de tilbudene som fins. Når noen benytter Altinns samiske løsning vil det føre til at andre får vite om den, og ønske å selv benytte løsningen. Når flere benytter den samiske løsningen vil andelen analfabeter gå ned, siden folk lærer seg offentlige ord og uttrykk på sitt eget språk.

Det er krav om 25 % nynorsk på offentlige nettsteder. Ved å tilby egne deler med nynorsk vil man gjøre mer enn dette. Selv om de fleste som har nynorsk som hovedmål

skjønner bokmål, kan det virke frustrerende at enkelte nettsteder skriver deler av tekstene på nynorsk for å oppfylle språklova.

3 Forutsetninger for oversetting

Dersom man skal utvikle flerspråklig støtte av et omfang som beskrevet i forrige kapittel, vil man møte en god del problemer. En løsning som støtter alle de største språkene i verden ville nok være meget nyttig, men dessverre får vi også da en god del problemer. Forskjellige alfabeter, utallige tegn og en god del dialekter per språk er noen av tingene som må tas hensyn til. For å forenkle oppgaven en del kan man derfor velge de språk som er mest hensiktsmessig å benytte i det systemet som skal utvikles. Vi tar derfor som utgangspunkt, at vi skal utvide Altinn med tre skriftspråk, nynorsk, samisk og engelsk. Tre språk som er noenlunde like, men som likevel har en del fundamentale forskjeller.

3.1 Tegn

På 1980-tallet da Internett begynte å ta form, og mens forgjengerne til mange av dagens it-systemer ble utviklet var det ikke mange som tenkte at bruksområdene skulle bli så store som de er i dag. I begynnelsen av utviklingen var de første tegnsettene basert på 7bits US-ASCII. Et system som støtter 128 tegn. Dette var nok til de engelske bokstavene og de mest brukte skilletegnene, men manglet støtte for bl.a. nasjonale norske tegn.

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	\$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(72	48	H	104	68	h
9	09	Horizontal tab	41	29)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	47	2F	/	79	4F	O	111	6F	o
16	10	Data link escape	48	30	0	80	50	P	112	70	p
17	11	Device control 1	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	50	32	2	82	52	R	114	72	r
19	13	Device control 3	51	33	3	83	53	S	115	73	s
20	14	Device control 4	52	34	4	84	54	T	116	74	t
21	15	Neg. acknowledge	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	54	36	6	86	56	V	118	76	v
23	17	End trans. block	55	37	7	87	57	W	119	77	w
24	18	Cancel	56	38	8	88	58	X	120	78	x
25	19	End of medium	57	39	9	89	59	Y	121	79	y
26	1A	Substitution	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	59	3B	;	91	5B	[123	7B	{
28	1C	File separator	60	3C	<	92	5C	\	124	7C	
29	1D	Group separator	61	3D	=	93	5D]	125	7D	}
30	1E	Record separator	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	63	3F	?	95	5F	_	127	7F	□

Tabell 2: ASCII -I

Da behovet etter hvert meldte seg, ble det utviklet mange forskjellige sett som skulle dekke det meste av språk. Et av de mest brukte seriene av slike tegnsett er ISO 8859 serien, som i dag er den mest brukte på Internett. Dette er en gruppe av standardiserte 8 bits flerspråklige tegnsett. En vesteuropeisk utgave er ISO 8859-1 som blant annet inneholder de særnorske bokstavene. Dette er i dag det mest brukte på norske websider. Dessverre støtter det ikke de samiske spesialtegnene. Det ble blant annet derfor designet et nytt sett, ISO 8859-10 eller Latin-6 som det også kalles. Dette støtter alle de nordiske språkene, også nordsamisk. Tegnsettet er støttet av de fleste nyere nettlesere.

Et annet godt alternativ er Unicode. Dette er egentlig en tabell som tilegner et tegn til et tall. Det finnes 8-bit, 16-bit og 32-bits utgaver, noe som gjør at man kan tilegne mange

flere tegn. Hvordan man skal vise tegnene kan by på problemer, siden så mye software er beregnet for å håndtere 8-bits tegnsett. Den interne logikken av 8-bits "legacy" systemer, tillater ofte kun åtte bits for hvert tegn, noe som gjør det umulig å bruke mer enn 256 tegn uten spesiell prosessering. En rekke mekanismer har blitt foreslått for å implementere Unicode. UTF-8 (Universal Transformation Format) ble utviklet nettopp for dette formålet. Dette er en metode for å gjøre om disse, for eksempel 16-bits, verdiene til 8 bits tegn. Metoden er støttet av alle nye nettlesere og kan med versjon 3.0 eller senere brukes for å vise samiske bokstaver.

Dessverre er det enda i dag i bruk en god del gamle nettlesere som ikke kan tolke verken Unicode eller Latin-6. For å løse dette er det laget en rekke "patcher" eller "addons" til gamle nettlesere. Dette er utvidelser av nettleserne som tillater dem å tolke slike tegn, og lenker til disse legges ofte ut til hjelp for personer som benytter slike nettlesere.

Det kan nevnes at mange nettsider bruker Windows-1252. Det er helt likt ISO 8859-1, bortsett fra noen ekstra tegn i området 128-159. Det ble også utviklet et lignende tegnsett for samisk, Windows-Sami-2. Det største problemet med denne løsningen var at det kun var Opera (6.0 og nyere) som støttet denne løsningen. Man måtte derfor også lage egne fonter for Windows-Sami-2 for å kunne se på slike nettsteder. På grunn av dette ble denne metoden ansett som en dårlig løsning, og kun antatt å være et midlertidig alternativ. Med Unicode 3.0 og ISO-8859-10 er det kommet bedre alternativer som burde benyttes.

Både UTF og ISO metoden er altså mulige løsninger på tegnproblematikken i vårt system. Fordelen med UTF-8 over Latin-6 er for det første at man slipper å bytte til annet tegnsett dersom man skal ha med enda flere språk. Metoden er altså mer fleksibel, siden den egentlig er basert på en 16-bits eller til og med 32-bits tegntabell. Likevel foretrekker mange ISO-måten. Et problem er dersom andre programmer enn nettlesere skal kunne kommunisere med systemet eller benytte dokumenter fra systemet. Programmer som ikke har støtte for valgt sett vil få ut en halvkryptisk tekst, som kan være lesbar for det menneskelige øye, men neppe kan tolkes av programmet. Spesialtegn vil her sannsynligvis vises som andre tegn eller ikke vises i det hele tatt.

Tidligere kunne ingen tegnsett holde nok tegn. Det trengs mange tegnsett for å dekke alle språkene i EU. Man kan også få problemer ved å benytte flere forskjellige tegnsett, siden to forskjellige tegnsett kan benytte samme nummer for ulike tegn, eller ulike nummer for samme tegn. Unicode tilbyr et unikt nummer for hvert tegn, uavhengig av plattform, program og språk.

Et alfabet er knyttet til et språk med tegnsett og sorteringskriterier. I Norge benytter vi dansk/norsk tegnsett, med spesialtegnene Æ, Ø og Å. Til tross for dette finner man ord som benytter tegn utenfor vårt alfabet. For å nevne et eksempel har vi ordet "gründer". Man kan ikke skrive "grynder", selv om det ville vært den naturlige direkte erstatningen av ordet. Det blir med andre ord benyttet et tegn som ikke finnes i det norske alfabetet, og det dukker opp et nytt problem, behandling av tegnsett utenfor språkets standard.

3.2 Fra tegn til informasjon

Et dokument består selvfølgelig ikke av bare vilkårlige tegn. En datamaskin må også vite hvordan den skal vise de forskjellige tegnene for at den skal kunne lage ord som leseren skjønner. Videre må ordene stå i riktig posisjon for at de skal få den riktige betydningen i setninger. Det er mulig å dele informasjonen som blir dannet av tegnene inn i 2 hovedgrupper:

- Kodbar informasjon er enten ord eller korte setninger med en fast betydning som kan kodes og muligens oversettes. For eksempel kan det være feltet *etternavn* i et skjema. Kodbar informasjon kan ofte være enklere å håndtere, siden de ofte har faste verdier/navn på alle språk. En datamaskin vil da forhåpentligvis selv ved hjelp av litt programmering kunne si at feltet *etternavn* skal ha navnet *surname* på engelsk. Kodbar informasjon er altså all informasjon som kan oversettes ved hjelp av en ”enkel” prosedyre i datamaskinen.
- Fritekst er vanlige tekster oppbygd av setninger, som for eksempel hele denne teksten. Disse tekstene er vanskelige å håndtere i og med at de innebærer alle problemer man kan komme bort i angående oversetting, sortering og tvetydigheter. Fritekst er all informasjon som ikke kan regnes som kodbar.

3.3 Sortering

I de fleste lands alfabeter er det også en standardisert måte å sortere tegnene på. Hvert språk har i utgangspunktet en egen måte å gjøre dette på, men det finnes heldigvis noen likheter. Dersom vi tar utgangspunkt i de nordiske språkene så ser vi at de fleste tegnene sorteres på samme måte, med noen få unntak. V og W sorteres likt på finsksvensk og man må se på neste bokstav, mens V sorteres før W på svensk. De svenske alternativene til Æ Ø Å sorteres Å Ä Ö. Hvorfor tegn sorteres på et bestemt sett har historiske årsaker, og er altså ikke alltid like enkelt å forklare eller sette regler til. Ofte er det enklere å se i en tabell hvordan dette gjøres.

Tabell 1 viser små og store bokstaver i det nordsamiske alfabet i sortert rekkefølge samt heksadesimal verdi i Unicode

A	Á	B	C	Č	D	Đ	E	F	G	H	I	J	K	L	M	N	Ŋ	O
41	C1	42	43	010C	44	0110	45	46	47	48	49	4A	4B	4C	4D	4E	014A	4F
a	á	b	c	č	d	đ	e	f	g	h	i	j	k	l	m	n	ŋ	o
61	E1	62	63	010D	64	0111	65	66	67	68	69	6A	6B	6C	6D	6E	014B	6F

P	(Q)	R	S	Š	T	Ʀ	U	V	(W)	(X)	(Y)	Z	Ž	Æ	Ä	Ø	Ö	Å
50	51	52	53	160	54	166	55	56	57	58	59	5A	17D	C6	C4	D8	D6	C5
p	(q)	r	s	š	t	Ʀ	u	v	(w)	(x)	(y)	z	ž	æ	ä	ø	ö	å
70	71	72	73	161	74	167	75	76	77	78	79	7A	17E	E6	E4	F8	F6	E5

Tabell 3: Små og store bokstaver i det nordsamiske alfabet i sortert rekkefølge. (Bokstaver som finnes i fremmedord er i parentes.) (Trosterud 2002: Samiske bokstaver) (Unicode)

I de fleste større systemer i dag vil mesteparten av data lagres i databaser. Når man henter informasjon ut av databasene vil mange hente den ut i sortert rekkefølge, siden disse som regel har gode innebygde sorteringsalgoritmer for dette. Dersom vi har et system som skal takle flere språk kan det være at den innebygde algoritmen sorterer på annet vis enn det vi vil ha svaret på. Dersom dette er tilfellet må systemet ha en egen algoritme som sorterer etter de regler som er satt av valgt språk.

Et annet litt kinkig problem vi har er ord med dobbel "A". Det ble tillatt å skrive "å" for "aa" i norsk i 1917 og ved rettskrivingsreformen i 1938 ble skrivemåten obligatorisk. Men mange foretrekker "aa" i egennavn, noe som er fullt tillatt. Dobbelt a skal vanligvis sorteres under "å", men dersom det uttales som "Aa", sorteres de som to "a"-er, altså først. De tilfellene der dobbelt "a" uttales som "a" utenom i egennavn er bøyningssendelser som for eksempel "en diva – divaa (divaen)" og "en primadonna – primadonnaa (primadonnaen)". I tilfellet Altinn er det nok heller egennavnene (fornavn, etternavn) der vi kan finne problemstillingen. Det er derfor svært vanskelig å lage noen regler for om disse uttales som "aa" eller "å". I Nederland er det for eksempel vanlig med dobbelt a i navn. Disse skal ikke uttales som å, og heller ikke sorteres som å. For eksempel Aalberg og Aalst.

Noe tilsvarende har vi med svensk "Ö" og tysk "Ö". Dette tegnet sorteres på svensk som beskrevet over, mens på tysk som "Oe". "Öland" vil i leksikon plasseres mellom "øl" og "Ølberg", mens "Böhmer" vil havne mellom "Boeck" og "Boel". En måte å løse dette på er å lagre ekstra informasjon sammen med ordene som sier hvordan disse skal sorteres. En annen måte er å implementere regler for hvordan ordene skal behandles. Dette fører igjen til at en sorteringsalgoritme vil ta lengre tid. En forenkling av sorteringen vil være å behandle alle ord med "dobbelt a" eller "Ö" likt.

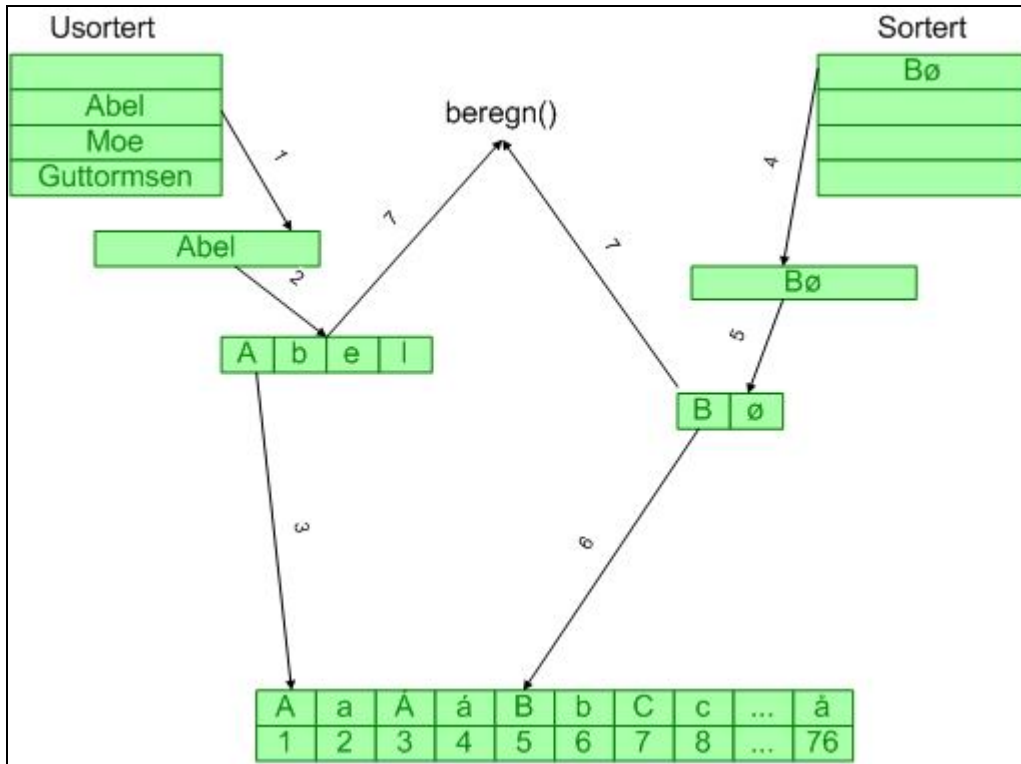
3.4 Kostnader ved sortering

Sortering er ofte noe av det som tar mest tid og ressurser i et system. Dersom man må utvikle en egen algoritme for sortering på spesielle språk, er det derfor viktig å gjøre en grundig analyse på hvordan dette kan gjøres mest mulig effektivt. Dersom for eksempel databasene har mange ledige ressurser, kan det være en god løsning å la disse gjøre mesteparten av jobben. Etter at resultatet da er noenlunde sortert kan man bruke en enkel algoritme som sorterer slikt materiale raskt. Denne metoden har flere fordeler. Man slipper å sortere mye utenfor databasen, man kan utvide en meget enkel algoritme og man vil få et raskt resultat.

Dersom data som skal sorteres, inneholder mange tegn som ikke kan sorteres av databasesystemet, er dette derimot en dårlig løsning. I dette tilfellet vil det være bedre å gjøre all sortering utenfor databasen. Denne måten burde også benyttes dersom databasen er flaskehalsen i systemet. Quick-sort algoritmen er en mye brukt og rask algoritme til å sortere større mengder data, som i utgangspunktet er helt usortert. Algoritmen vil derfor være ganske egnet til denne bruken.

Insert-sort kan også være egnet dersom man legger til et og et element inn i databasen. Metoden fungerer også best når man har et fast kriterium å sortere på. Man vil da kunne plassere elementene på riktig sortert plass ved innsetning. Noe man må passe på er at alle delene av databasen blir gjort på samme måte, slik at man ikke får usortert resultat når man henter ut forskjellige deler av databasen og sammenligner disse.

Figur 1 beskriver hvordan en slik algoritme kunne fungere i praksis, og hvordan den benytter en sorteringstabell. Algoritmen består av to tabeller, en sortert og en usortert. I tillegg benytter den en tredje tabell, sorteringstabellen der den henter ut verdiene til bokstaver i ordene. Før man gjør noe som helst sortering, vil den sorterte tabellen være tom og den usorterte være full. Algoritmen tar så ut det første elementet fra den usorterte tabellen (1). Deler så inn i tegn (2) og henter verdien av første tegn fra sorteringstabellen (3). Gjør så det samme for den sorterte tabellen (4,5 og 6). Dersom det ikke finnes noe element i tabellen, plasseres ganske enkelt ordet vi skal sortere inn fremst i den sorterte tabellen. Neste steg består av å sammenligne verdiene, for å bestemme om man må se på neste tegn, om man kan plassere elementet, eller om man må se på neste ord i den sorterte tabellen. ”*Introductions to Algorithms*”, av Cormen, Leieron, Rivest og Stein gir grundige analyser av blant annet sorteringsalgoritmer og effektivitet.



Figur 3: Inset-sort.

Uansett hvilken sorteringsmekanisme som benyttes må den vite rekkefølgen den skal sortere tekststrengen i. En måte å gjøre dette på er å tilegne hvert tegn en verdi til alle språk den skal sorteres etter. På denne måten så kan man lage forskjellige sorteringsregler etter hvilket språk man skal sortere på. Man kan dermed sortere "Ö" forskjellig på svensk og tysk. Metoden vil nok være ganske ressurskrevende, og i systemer som er helt avhengig av raske svar kan det være hensiktsmessig å forenkle denne. En forenkling kan være å sortere likt for alle språk, og bare sortere på tegn. Dermed vil svensk og tysk "Ö" sorteres likt. Spørsmålet blir da om man vil velge den svenske måten, eller den tyske.

Ved å studere Unicode verdiene i tabell 1 nærmere ser man at disse ikke er gunstig. De norske tegnene står nesten i stigende rekkefølge etter slik vi ville sortert dem, men med unntak av "Æ", "Ø" og "Å".

For å tilegne hvert tegn en verdi kan man bruke en tabell. Her dukker det imidlertid opp noen problemer som må løses. Som nevnt kan man lage en tabell, eventuelt en kolonne for hvert språk. For de fleste tegn er dette ganske enkelt, men noen unntak er det dessverre. Siden "Ö" sorteres som "OE" på tysk er det ikke bare å sette en verdi på dette tegnet. Ofte kan det være enklere å tilpasse tabellene etter sorteringsalgoritmen, enn å tilpasse algoritmen etter tabellene.

3.5 Verktøy

3.5.1 Nettlesere

Nettleseren er et program som brukes for å hente opp og vise fram informasjon lagret i form av dokumenter på Internett. Det er i dag mange forskjellige nettlesere i bruk. Microsofts Internet Explorer (IE), Netscape, norske Opera og Firefox/Mozilla er vel kjent for de fleste, men det finnes en rekke andre som er gode til sitt bruk. Blant annet finnes det rene tekstbaserte nettlesere.

Det er de siste årene også utviklet en egen protokoll for mobiltelefoner, kalt WAP, Wireless Application Protocol. Protokollen gjør at man får tilgang på ressurser fra Internett gjennom mobiltelefonen. WAP nettsteder er som regel bygd opp på en slik måte at det blir minst mulig å laste ned for nettleseren. Dette er fordi mobiltelefoner og andre håndholdte gjenstander som regel har lite minne, lite lagringsplass og liten skjerm. Selv om WAP støtter HTML og XML er WML, en XML applikasjon, spesielt beregnet for små skjermer og en veis navigasjon uten keyboard. WAP støtter også WMLScript som er et scriptspråk ikke ulikt JavaScript, men som krever mye mindre ressurser.

Selv om mange nettlesere i utgangspunktet kan virke ganske like, i måten de presenterer data, er det likevel en del forskjeller. Opera itererer for eksempel over tabeller i annen rekkefølge enn IE og Firefox. De siste itererer i den rekkefølgen tabellen ble initialisert, mens Opera itererer etter den sorterte rekkefølgen av indeksen. Mange kritiserer også IE for å pynte på dokumentene. Nettleseren stiller svakere krav til den som lager dokumentene, og retter automatisk opp en del småfeil som blir gjort i kodingen. Mange foretrekker IE nettopp på grunn av at den pynter på dokumentene, mens andre mener at IE er årsaken til at html kodingen er så lite standardisert i dag.

Nettleseren skal tolke informasjonen fra serveren, vise den til bruker, tolke input fra bruker og levere til server. Dette betyr at nettleseren må kjenne til de tegnssett og teknikker som serveren benytter. Ikke alle norske nettlesere kan presentere en japansk side uten å installere en "add-on" først. Nettleseren må ha installert det japanske tegnsettet/tegnsettene før de kan vises til bruker. Ikke alle nettlesere har funksjoner for å laste ned tilleggsfonter.

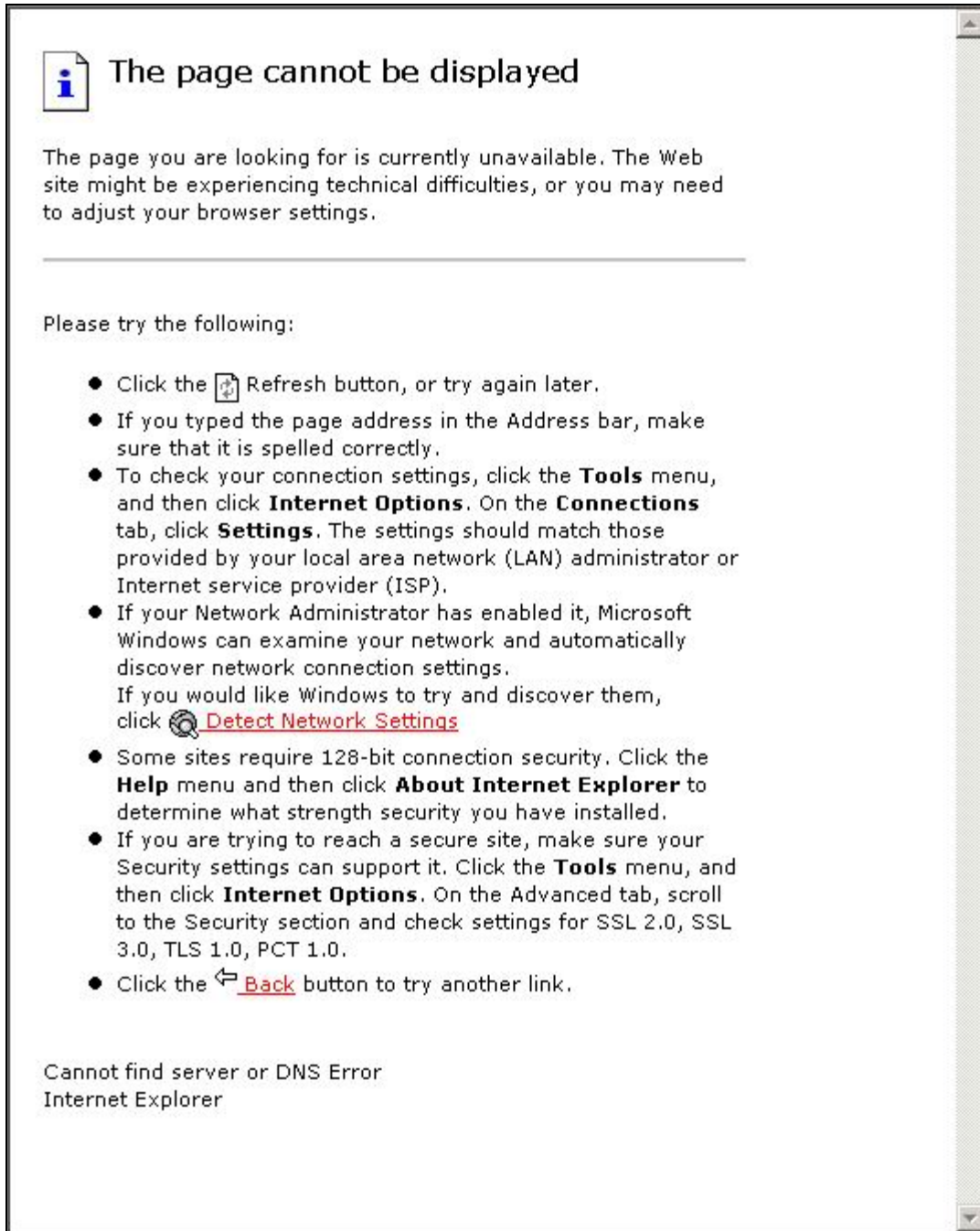
3.5.2 Domenenavn

Registreringstjenesten Norid åpnet i mars 2004 for å registrere domenenavn med æ, ø, å og andre "avvikende" tegn. Interessen var stor. På under tre måneder ble det søkt om 8.664 nye domener med enten æ, ø eller å. I tillegg søkte 144 om domener med samiske tegn eller aksenter i navnet.

"Det har ikke vært helt problemfritt ettersom det ikke er alle programmer som støtter æ, ø og å", sa Norid-sjef Hilde M til nettavisen Digi i juni 2004. Thunem anbefaler derfor at

brukerne opererer med flere alternativer. Bærum kommune har for eksempel fått tildelt domenenavn med æ, men bør fortsatt beholde Baerum-domenet.

Det største problemet er støtten fra nettlesere. Verken Internet Explorer eller Opera 7.x støtter i utgangspunktet domenenavn med særnordiske tegn. Når man forsøker med IE, får man opp følgende feilmelding:



Figur 4: <http://www.språkrådet.no> i IE 6.0 uten tilleggspakke

Feilmeldingen forklarer lite om hva problemet egentlig er, og ingenting om at man faktisk kan laste ned oppdatering som gjør at nettleseren tillater dette. Oppdateringer finnes for Mozilla 1.4 eller nyere, Netscape 7.1 eller nyere og Opera 7.20 eller nyere. Siste utgave av Opera, 8.0, har innebygd støtte for dette, mens VeriSign har utviklet løsninger for Microsoft produktene, IE 5.0 eller nyere, og Outlook.

Norske tegn i domenenavn var etterspurt, men lot seg ikke realisere fordi de ikke er generiske nok. Selv om nettleserstøtten skulle bedres med årene, vil problemene med særtegn i tastatur og lignende fremdeles være der.

3.6 Bruk av Unicode

En god del gamle nettlesere er enda i bruk. Mange av disse kom før Unicode og har heller ikke muligheter for å tilpasse seg det, men også med nye de som bruker nye nettlesere kan få problemer.

Som nevnt må man installere et tillegg for å vise japanske tegn i en norsk nettleser. Dette gjelder også når man benytter Unicode. Det man må legge til her, er fonter/tegn til verdiene i Unicode tabellen. Ideelt så vil man installere fonter for de sidene man trenger, og i tillegg en full Unicode font som backup. For Windows 2000 kan man legge til ønskede språk på følgende måte:

Start > Settings > Control Panel > Regional Options.

Under "General" tabulatoren kan man så legge til ønskede språk. Dette legger til fonter og systemstøtte.

I Mac OS X har Safari støtte for Unicode, mens OmniWeb direkte støtter Unicode tegninger. OmniWeb støtter ikke alt av Unicode, men kan dra nytte av fonter fra Windows dersom de er installert på riktig måte. OS 9.x har ingen nettlesere som støtter direkte tegning av Unicode. Alle bruker Apple Language Kits og WorldScript i forskjellig grad for å støtte Unicode og internasjonal tekst. Forskjellige Language Kits kan installeres gjennom Mac OS 9.x installasjons CD.

Nye fonter med fast bredde for X11/Unix brukere kan lastes ned, men tidligere versjoner av disse er automatisk installert med XFree86 4.0 X serveren, som er den mest brukte under Linux. Mozilla og Netscape 6 kan benytte seg av disse fontene direkte.

Det er tre ofte brukte metoder for å representere tegn som ikke finnes på tastaturet i dokumenter. Dette kan være matematiske tegn eller tegn fra andre språk.

1. 252 tegn kan inkluderes i et HTML dokument ved å skrive inn et symbolsk navn mellom en ampersand og et semikolon, for eksempel ø for bokstaven ø. Tegnreferansene, som dette er, skal vises uavhengig av dokumentets tegnsett, og bør virke med alle HTML filer.

2. Numerisk representasjon er en annen måte å sette inn tegn på. Her skriver man i stedet for navnet til tegnet inn en desimal numerisk referanse mellom ampersanden og semikolonet. Må også vise at det er en numerisk referanse ved å legge til et # før referansen. For eksempel vil `—`; vise en em dash (—).
3. Heksadesimale nummer er også et alternativ. Em dash tegnet vil da være representert med `—`; Der x viser at det opereres med heksadesimale tall.

Unicode fonter tillater komplette tegnssett for flere språk i en enkel fontfil, men de trenger ikke inneholde alle Unicode tegnene. Fonter for et spesifikt språk gir ofte resultater som er mer akseptable, for personer som snakker språkene, enn fonter som forsøker å dekke mange språk. Editorer og nettlesere bør være i stand til å gjøre det mulig for mer enn en font.

Windows har en rekke fonter til bruk med Unicode. Noen av disse er for spesifikke språk, mens andre, som Arial Unicode MS og Code2000, dekker flere språk og skrift. Mac OS X 10 kan benytte fonter ment for Windows, og kommer med en rekke Mac Unicode fonter som tillater skrifter å bli manipulert og vist på skjerm.

Selv om det normalt ikke er nødvendig, kan man bruke stiler for å spesifisere foretrukket font for tekster i et bestemt språk, ved å definere en klasse i et stylesheet.

```
<style type="text/css">
.thai {font-family:"Cordia New",Ayuthaya,Tahoma,"Arial Unicode MS";}
</style>
```

Man kan så benytte stilen i en HTML tag i `<body>`, og også spesifisere et språk til en del av teksten.

```
Normal tekst<span class="thai" lang="th">Thai text</span> Normal tekst.
```

Når nettleser støtten for flerspråklighet blir bedre, vil spesifisering av språk hjelpe med å tilby bedre språkspesifikk visning av diakritiske tegn, kombinasjoner av tegn, tegnsetting og orddeling.

Ingen av HML editorene eller ordprosessorene i Mac OS 9, kan benytte Unicode TrueType, selv om operativsystemet støtter det. I stedet benytter de Language Kits som benytter Apples tegnssett for å ta skrive, vise og skrive ut spesialtegn, for så å konvertere til Unicode, når en fil er lagret, med UTF-8 koding.

De tre største nettleserne for Windows inkluderer støtte for Unicode TrueType fontene, og kan derfor vise omtrent hvilket som helst tegn du vil finne på en webside. Mozilla har sannsynligvis den beste støtten siden den kan blande tegn fra flere fonter for å vise hvilket som helst tegn. Internet Explorer og Opera 7 er ikke fullt så gode på å blande tegn fra flere fonter, mens nettleserne for Mac OS 9 ikke er så gode som de for Windows, siden de kun kan vise tegn støttet av en allerede installert Language Kit.

4 Oversetting

Tekster må oversettes fra et språk til et annet. Mange har forsøkt på å lage automatiske oversettelser, med variabel suksess. Kapitlet beskriver forskjellene mellom fritekst og kodbar informasjon, problemene med oversetting av fritekst, og løsninger som kan benyttes ved kodbar informasjon.

4.1 Automatisk oversetting av fritekst

Den automatiske metoden er særdeles komplisert. Dersom det ikke finnes et program på markedet for dette, må man utvikle kode som oversetter fra standard språk til de ønskede språkene. Dette er for fullstendige setninger en meget komplisert prosess. Programmet må, i tillegg til en fullstendig ordbok, inneholde grammatiske regler. For språk som tilhører samme språkgruppe, som norsk og svensk, vil disse reglene være noenlunde like, mens de for norsk og samisk vil være temmelig ulike.

Oversetting av enkle ord er en del enklere enn fullstendige setninger, men selv dette er meget avansert. Mange ord har samme skrivemåte, men helt forskjellig betydning. For å kunne løse dette trengs det tilleggsinformasjon i ordboken som gir betydningen av ordene. Oppbyggingen av en slik løsning opp fra grunnen vil kreve ekstremt mye ressurser, både personer og tid. Testing bør gjennomføres over lengre tid og mot store og forskjellige grupper mennesker. Fordelen er at når dette er utviklet kan man spare ressurser på oversetting av tekster, i tillegg til at man har et meget ettertraktet produkt.

Det finnes en del programmer som kan gjøre oversettingen allerede på markedet. Et av de bedre er kanskje Babel Fish som oversetter hele nettsider automatisk fra et valgt språk til et annet. Et annet er InterTran som faktisk har norsk som et av språkene. Dessverre er resultatene man får langt fra gode. ”Dette er en enkel test av oversettelsesprogrammet til InterTran.” ble til ” This am a efrem test at compiling program at InterTran.” Etter dette er gjort kan man, i programmet, forandre en del av ordene og sette inn nye ord. Dette er vel for å prøve å lage setninger med noe mening i, men her går nok vinningen opp i spinningen.

Til både Babel Fish og InterTrans forvar er det ingen drøy påstand at et helautomatisk oversettelsesprogram neppe er realiserbart. For å vise kompleksiteten i et slikt program, kan man bruke en så enkel setning som ”slå på tv-en”. Setningen på engelsk kunne blitt enten ”turn on the tv” eller ”hit the tv”. For å få en korrekt oversettelse er man her avhengig av å vite hvilken sammenheng setningen står i. Kanskje med nok kunnskap om resten av teksten, kan programmet så klare å oversette setningen. Hva så om man ikke kan oppdrive denne kunnskapen? Sett at den siste setningen i en novelle var ”og i sinne slo han på tv-en”. Det er godt mulig at forfatteren av denne setningen hadde ment å

plassere dette ordspillet her, noe en automatisk oversettelse til engelsk ville hatt store vanskeligheter med å beholde.

4.1.1 Nyno

Det norske firmaet Nynodata har lagd et program som oversetter fra bokmål fritekst til nynorsk. Metoden programmet, Nyno, bruker er ganske lik metoden til InterTran.

”Nyno er laga for å lette arbeidet med sambruk av bokmål og nynorsk. Nyno tek utgangspunkt i ein tekst på bokmål og skriv han om, delvis automatisk og delvis i samarbeid med brukaren. Programmet tek omsyn til vanlege grammatikkreglar og syntaksreglar for nynorsk innafor gjeldande rettskriving. Læreboknormalen ligg som standard, men det er mogleg å bruke sideformer. Normalt skal brukaren godkjenne setningane etter kvart som dei blir omsette. Alternativt kan ein køyre heile teksten igjennom og korrekturlese etterpå.” (Nyno)

Dette er hva Nynodata sier selv om sitt program. Nyno har vært tilgjengelig siden 1995 i ulike versjoner. Programmet har til nå mesteparten av brukerne sine i offentlig sektor. Over halvparten av departementene bruker det, og 5 har kjøpt det for alle ansatte. Andre som kan nevnes er statens forvaltningstjeneste, Stortingsbiblioteket, og Kommunenes Sentralforbund. Det bør også nevnes at etter at Microsoft brukte programmet til oversetting av Office 2004 til nynorsk hevdet de at denne jobben ville vært umulig uten hjelp av Nyno. Denne kommentaren virker kanskje litt drøy på nordmenn med kjennskap til begge språk, men betyr kanskje at programmet sparte dem for så mye tid og penger at de faktisk kunne tjene på å oversette til nynorsk.

Programmet passer også godt inn i feltløsningen som beskrives senere I oppgaven. Ved å benytte Nyno til selve oversettingen fra bokmål til nynorsk, krever det bare at man har noe kjennskap til nynorsk. Oversettingen er en god del enklere, og går mye raskere enn om man skulle gjøre den helt for hånd. Selv personer med nynorsk som hovedmål skal vel ha problemer med å oversette raskere enn Nyno.

Bokmål og nynorsk kan beskrives som forskjellige presentasjonsformer av samme språk. Med dette menes at det kun er ørsmå forskjeller på skriveformen, og setningsoppbyggingen er stort sett helt lik, i hvert fall om man benytter den løseste formen for nynorsk. Dette er vel det som i hovedsak gjør Nyno så mye mer vellykket, og enkelt enn for eksempel InterTran. De valgene man må gjøre er stort sett de få unntakene, som for eksempel genitivs s i bokmål.

4.2 Oversetting for hånd

I stedet for å ha programmer som gjør oversettingen er det selvsagt også mulig å gjøre oversettingen for hånd. Man oversetter da gjerne fra et språk til de andre, legger tekstene

i forskjellige filer. På en nettside vil man da for eksempel skille de forskjellige språkene med ulike adresser. Personer med god kjennskap til, både det opprinnelige språket og det språket som skal oversettes til, må stå for selve oversettingen. I tillegg må man ha folk som kan stå for oversettingen når det blir gjort forandringer i systemet.

Metoden er en god del enklere å implementere enn automatisk oversetting, men kravet om personressurser til oversetting også etter at systemet er implementert kan være en merkbar utgift. Metoden krever litt ekstra lagringskapasitet, ved at lagringsvolumet for alle tekster blir tilnærmet multiplisert med antall språk. Økningen i aksestid vil for det meste bestå i søketid etter riktig tekst.

Metoden er gunstig i en rekke tilfeller. Med få språk kan metoden være helt akseptabel. Dersom man tillater, eller ønsker at språkversjonene skal være forskjellige kan dette være den eneste måten å gjøre det. Utgavene kan da regnes som to forskjellige nettsted, med noe likt innhold. Ofte vil man da også ha forskjellige personer ansvarlige for de forskjellige utgavene.

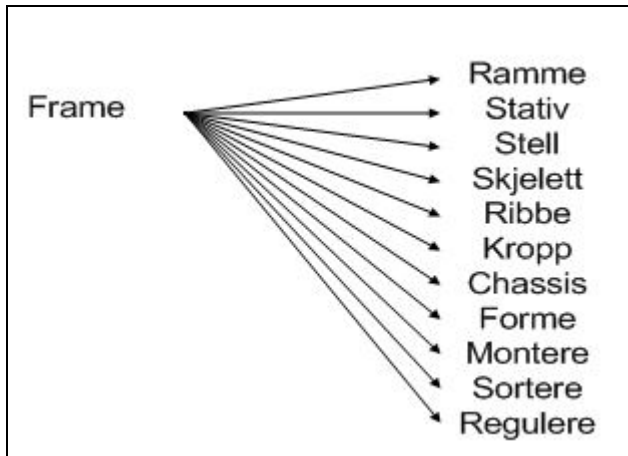
4.3 Oversetting av kodbar informasjon - ordbokmetoden

Som tidligere nevnt er det ofte mulig å gjøre oversettelsen av kodbar informasjon enklere enn ved fritekst. Hvis vi tar utgangspunkt i EU og den økende skjematrafikken mellom ulike EU/EØS land, så finner vi mest sannsynlig en del gjentakende felter, for eksempel dato. Ved å tilegne alle gjentakende felter en bestemt verdi kan man ut i fra en database, som programmet må ha tilgang til, kunne oversette direkte. Denne metoden vil innebære et databasekall på alle verdiene, for å kunne oversette, men vil være en del enklere å implementere enn automatisk oversettelse av fritekst. Denne metoden er å sammenligne med en slags automatisk ordbok, med det tilleggsargumentet at den må velge riktig betydning av ordene.

Det er en del vanskeligheter her også. Den største utfordringen er kanskje å lage denne databasen med alle verdiene.

- For det første vil et skjema som inneholder rader for alle land i EU/EØS (19 språk i skrivende øyeblikk), og kolonner for alle ord/uttrykk som blir brukt bli et meget stort skjema. En måte å krympe skjemaet en del kan være ved at hvert land har et skjema med rader for de språkene de behøver, og kun det. En annen løsning vil jo være å dele inn skjemaer i kategorier, for eksempel fiskeri og landbruk, slik at man får mindre kolonner. Eventuelt kan disse kombineres. Dette er vel ganske vanlige databaseproblemer, og bør vel løses av databasedesignerne.
- Alle ord må oversettes av personer før løsningen kan tas i bruk.
- Noen ord kan bety flere ting på forskjellige språk. En fallgrube kan derfor være hvis man kun tar et språk som utgangspunkt i utviklingen av databasen. Dersom man da har flere betydninger av et ord må man være forsiktige at man tilegner de forskjellige betydningene forskjellige verdier. Med andre ord er det kanskje

betydningene av felter man bør gi verdier, ikke ordene/setningene i seg selv. Problemet blir da at man kan få vanskeligheter med å si hvilket felt som er det korrekte siden man har flere ord/setninger med forskjellig verdi. En eller flere tilleggsopplysninger må altså til.



Figur 5: Ord kan ha flere betydninger. Her noen norske oversettelinger av det engelske ordet frame.

Kode	Bokmål	Nynorsk	Samisk	Engelsk
100100	Ramme	Ramme	Rámma	Frame
100101	Stativ	Stativ	Holga	Frame

Tabell 4: Ved å tilegne ord en verdi så kan man skille mellom betydningen av samme ord. Her skilles betydningen av det engelske ordet frame ved å gi den to verdier, som tilsvarer ordene ramme og stativ på norsk.

4.4 Oversetting av kodbar informasjon – feltløsning

For mindre systemer med kun et begrenset antall felter, kan det av disse grunnene være en god ide å forenkle systemet litt. Et eksempel kan være for eksempel en e-post. Denne kan bestå av 5 felter. ”Fra”, ”til”, ”subjekt”, ”innhold” og ”send” knappen. Her kan vi gi alle feltene et nummer, for eksempel fra en til fem. Et medfølgende skjema gir feltene en verdi på de språk som trengs. Et eksempel på et slikt skjema er tabell 2. Etter at systemet får vite hvilket språk som skal brukes, kan det da hente ut og vise feltene på riktig språk. Ved å tillegge hvert felt en verdi i stedet for hvert ord, slipper vi å bekymre oss for ordenes betydning. Vi slipper dermed også å bekymre oss for om et ord kan bety flere ting på forskjellige språk. Fremdeles må man oversette feltene for å lage skjemaet, og det er stor fare for at vi får gjentakende felter. For mindre systemer vil denne løsningen likevel være mer aktuell siden den er enklere å implementere enn ordboken beskrevet over.

Figur 6: Enkel e-post, her på bokmål

Feltnr	1	2	3	4	5
Bokmål	Fra	Til	Emne	Innhold	Send
Nynorsk	Frå	Til	Emne	Innhald	Send
Samisk	Geas	Geasa	Fáttá	Sisdoallu	Sádde
Engelsk	From	To	Subject	Content	Send

Tabell 5: Mulig skjema for flerspråklig løsning av enkel e-post. I tillegg til feltene tatt med i tabellen, er også feltene man skriver i viktige for løsningen. Informasjonen der må kunne tydes av mottaker. Det er derfor mer egnet med numeriske verdier og valgli ster enn fritekst her. Dermed blir e-posten et dårlig eksempel.

For å bestemme seg for hvilken av disse to metodene man skal bruke er det flere ting man må tenke på:

- Antall felter
- Antall språk man skal oversette til
- Hvor forskjellige språkene er
- Hvor stor forskjellene på feltene er
- Hvor ofte går det samme feltet igjen

Det er utfordrende å gi svar på når man burde vurdere å lage en ordbok i stedet for en feltløsning, men ved å vurdere disse punktene kan man komme en del nærmere et svar. Som nevnt tidligere kan kompleksiteten av en ordbok bli særdeles høy. Dersom språkene som skal benyttes er svært like, som for eksempel bokmål og nynorsk, vil kompleksiteten

naturligvis bli minsket. Med mange like felt, med samme betydning, kan også denne løsningen virke mer tiltalende.

Siden hastigheten på datamaskinene og databasesystemene i dag er så stor, og fremdeles øker kraftig, skal nok antall felter og antall språk være særdeles store for at dette skal påvirke en beslutning om å velge ordbok over feltløsningen. Her må det en grundig undersøkelse av tidsbruk til for å finne ut hvor mye tid man kan spare på det ene over det andre.

En Western Digital S-ATA 120GiB disk har oppgitt søketid 9,20 ms, og testet lesetid til omtrent 48MB/s. Tiden det tar å hente ut en fil fra disk er aksesstid + lesetid. Aksesstid er tiden det tar å finne frem til filen. Søketid er tiden det tar å søke frem til rett spor på disken. Rotasjonsforsinkelsen er tiden det tar før platen snurrer rundt til det stedet man skal lese fra. I gjennomsnitt en halv omdreining. På en 7200rpm disk er dette alltid 4,17 ms.

Aksesstid = søketid + rotasjonsforsinkelse

Aksesstid = 9,20ms + 4,17ms

Aksesstid = 13,4ms

10.000 felt på gjennomsnitt 10 tegn, og et språk er ca 100kB. Det tar med andre ord

100kB / 48000kB/s = 2,08ms å lese dette

Det tar altså ca 15,6ms å lese 10.000 felter av ti tegn på et språk.

Lesetiden ved 100.000 felt vil være:

1000kB / 48000kB/s = 20,8ms

Som totalt blir 34,2ms inkludert aksesstid.

Lesetid ved 1.000.000 felt:

10000kB/48000kB/s = 208ms

Som totalt blir 221ms

En test program skrevet i Java, for kun å lese et slikt skjema uten noe mer prosessering, fikk resultatene i tabell 3. Målingene viser hvor lang tid det tar å lese en tekstfil av denne størrelsen på en virkelig maskin. Vi ser at målingene skiller litt fra utregningen over. Det er flere årsaker til dette. Flere programmer ble kjørt samtidig som målingene. Størrelsene er avrundet. Målingene er også gjort på en personlig datamaskin, der data på disk ikke er plassert optimalt. Dermed får er det sannsynligvis mer søketid enn i eksemplet over. Et mer avansert system ville sannsynligvis ikke lest alle felter for å finne de opplysninger det trengte. Et større system ville kanskje heller ikke brukt en enkel tekstfil for å lagre disse opplysningene. Tabellen viser likevel at det finnes begrensninger for hvor store skjema vi kan ha. Den viser også at det må størrelsen bør tas i betraktning når vi har med meget store systemer og vurderer ordbok mot feltløsningen. En forsinkelse på 16ms er så

liten at man den ikke ville være merkbar for de fleste brukere. En forsinkelse på 62ms er heller ikke merkbart stor. 500ms er derimot såpass mye at til og med brukeren kan merke dette. Uansett så ser man av målingene og utregningene at det må til meget store antall felt for at belastningen skal bli så stor at man må vurdere ordbok over feltløsning.

Antall felt	Størrelse på fil i kB	Gjennomsnittlig lesetid ms
10.000	100	16
100.000	1000	62
1.000.000	10000	500

Tabell 6: Hvert felt består av 10 tegn. Størrelsen er avrundet for lettere lesning. Filen er en tekstfil i Windows 2000 NTFS filsystem og målingene er gjort på en AMD64 3200, 1024MB RAM.

Alt etter hvordan man implementerer systemet, så kan det være aktuelt å overføre felttabellene til brukermaskinen. Overføringer over Internett tar enda lengre tid enn harddiskoppslag. En bruker med modem med overføringshastighet på 56kb/s vil bruke ca $100\text{kB} * 8\text{B/b} / 56\text{kb/s} = 14\text{s}$ på å overføre en felttabell med 10.000 felt over Internett. Velges en slik løsning bør man gjøre det man kan for å gjøre overføringene så sjeldne og så små som mulig.

4.5 Bruksområder for skjematisk oversettelse

Det er ikke bare det offentlige som kan ha nytte av en god løsning for skjematisk oversettelse. Mange private bedrifter kan nyte godt av en slik løsning. Et eksempel er varehandelen. Løsningen kan gjøre det enklere for nettbaserte butikker å utvide markedet sitt ut av landet. Butikker kunne hatt enkle elektroniske løsninger for å vise informasjon om produktene sine på mange flere språk enn det de har i dag. Butikker på områder med mange som ikke har norsk som hovedmål, og som kanskje har mange forskjellige språk å forholde seg til, kan ha beskrivelser av produktene. Skjema med beskrivelser på ønsket språk kan sendes ut av produsenten av produktet eller av en egen utvikler av slike skjema. Kunden får på denne måten en god beskrivelse av varen på et språk han kan forstå.

Pass, førerkort, studentkort, transportdokumenter, billetter, E111 (europeiske trygdekort), helsekort, fakturaer, vitnemål, skatteoppgaver, firmaattester, innholdsdeklarasjoner for matvarer er eksempler på hvor skjematisk oversettelse kan benyttes.

Et litt morsomt eksempel hvor man kan benytte feltløsningen til oversetting er Bibelen. Den er inndelt i vers, eller felt om du vil. Versene er alle oversatte til mange språk, men beholdt versformen. Det vil altså være ganske enkelt å legge til verdier til hvert felt for så å bruke feltløsningene og oversette til de språkene som er ønsket. Skjemaene er så si ferdige, bare å implementere med verdier og vi har en flerspråklig interaktiv utgave av Bibelen.

Fører kort er allerede standardiserte igjennom EU og benytter en slags feltløsning. De aller fleste feltene trenger ikke oversetting, men tre felt har det. Dette er navnet på landet, Norge, navnet på kortet, fører kort, og beskrivelsen av feltene. Alle personopplysninger og fører kortopplysninger er standardiserte og skal kunne tolkes i alle EU land. Kortet kan dermed gi tillatelse til å føre bil i alle EU land. Dersom man flytter til for eksempel Norge og har et slikt kort, kan man altså fortsette å benytte dette. Dersom man derimot har fått utstedt fører kort fra et land utenfor avtalen, må dette fornyes senest innen et år.



Figur 7: Forside av fører kort, EØS-modell. På høyre side skimter man så vidt fører kort skrevet på de forskjellige språkene. I midten, til høyre for bildet, ser man de feltene med personopplysninger.

5 Praktisk eksempel – Altinn

For å vise hvordan problemene kan løses ved bruk av feltløsningen kan vi bruke Altinn som eksempel, og gå igjennom de viktigste punktene når man skal gjennomføre et slikt prosjekt.

5.1 Kravspesifikasjon

Før man begynner å gjøre om et større eksisterende system om til en flerspråklig løsning må man altså vurdere en rekke valg. Vil derfor prøve å gjøre noen av disse vurderingene som en slags kravspesifikasjon for Altinn.

5.1.1 Språk

Det første man kanskje vil tenke på er hvilke språk som er aktuelle for systemet. Bokmål finnes fra før av. Ved å ta med nynorsk og samisk vil man få dekket språkloven og alle de offisielle skriftspråkene i Norge. I tillegg vil i det minste engelsk være et nyttig språk å ta med. Skatteetaten har allerede ytt et ønske om å få på plass en engelsk løsning. Mange vil kanskje anse dette som en enkel del av oppgaven. Engelsk har ingen sære tegn som lager problemer i oversetting eller sortering. Mange har gode kunnskaper i engelsk, så for å oversette feltene til engelsk trenger man ikke hente inn folk utenfra. Det siste er ikke helt korrekt. Norske skatteord byr på noen spesielle utfordringer. Siden de fleste land har utviklet sine egne måter å beregne skatt på, har man også utviklet egne uttrykk for å beskrive reglene på. Dette gjelder også i Norge. Både ordet og fenomenet toppskatt er helt ukjent i for eksempel USA. Det å beskrive norske skatteuttrykk og lovverk med engelsk til en amerikaner som aldri har hørt om toppskatt kan mildt sagt være utfordrende. For at en slik løsning skal kunne fungere tilfredsstillende må man altså ha en mye større grad av beskrivelser og hjelp i den engelske løsningen.

Det bør også vurderes om systemet skal kunne utvides med enda flere språk i fremtiden. Som nevnt tidligere er det blitt en stor gruppe russisktalende i Finnmark. En gruppe som kanskje etter hvert vil komme med ønske om russisk løsning. Kanskje kommer Norge med i EU og ser fordeler med å utvide med en rekke europeiske språk. Det er vanskelig å forutse fremtiden så det beste er nok å lage en løsning som kan være lett å utvide, med flere språk, ved behov.

Velger altså å lage en fleksibel løsning med stor vekt på utvidbarhet, og med de fire først valgte språkene:

- Bokmål
- Nynorsk
- Samisk
- Engelsk

Altinn avtalen har følgende punkter:

- Krav til språk (Portal)
 - Applikasjonen skal være fleksibel i forhold til bruk av ulike språkformer og tegnsett. Bokmål, nynorsk, samisk og engelsk er språk som løsningen skal håndtere.
- Krav til hjelpefunksjoner (Skjema)
 - Det skal være en online hjelpefunksjon tilpasset ulike språkversjoner. Hjelpe-funksjonen skal være kontekst avhengig. Hjelpefunksjon skal være tilgjengelig både på skjemanivå og på feltnivå. Innlegging av hjelpetekster må kunne angis ved oppbygging av skjema.

5.1.2 Brukerkrav

Altinn avtalen sier også følgende om brukerkrav:

- Det er viktig for fellesløsninger som Altinn, at innføring av flerspråklighet skjer koordinert fra offentlige etater
 - som for eksempel innføring av samisk
- En felles offentlig portal som tilbyr samisk bare i portalen, eller bare i enkelte skjema vil være svært forvirrende for de brukere som ønsker å bruke samisk i slike sammenhenger

Det siste punktet sier noe om dette, men det er viktig at også resten av løsningen er konsistent på tvers av flerspråkligheten. Det er mer enn ønskelig at det man holder på med i bokmålsløsningen finnes igjen på samme sted i den samiske. Dette gjør også at det blir enklere å drive med ”kundeservice”, og å lage hjelpefunksjoner og beskrivelser.

5.1.3 Tegnsett

Etter at man har funnet ut hvilke språk som er ønskelig, er det nødvendig å finne et tegnsett som passer for språkene man skal utvide med. Som nevnt tidligere er det to av de mest brukte tegnsettene som passer meget godt for de valgte språkene, ISO 8859-10 og Unicode. Sistnevnte gjør det også mulig å utvide med flere språk uten å bytte tegnsett mellom språk, og vil dermed gjøre løsningen mer fleksibel.

Et problem her er at skjemamotoren i Altinn er Adobe Form Server som ikke foreløpig støtter samiske tegn. Sommeren 2006 planlegger Altinn å bytte til versjon 6.0, som skal støtte samiske tegn.

5.1.4 Sortering

Som nevnt tidligere kan sortering være ganske komplisert når det skal inkludere mange forskjellige språk. Vi må altså lage regler for hvordan dette skal gjøres, og sortere ut fra dette. Her må det tas valg om sorteringen skal skje etter norske regler tilpasset til de spesielle tegnene i samisk, og eventuelt andre språk etter hvert, eller om man skal sortere forskjellig for hvilket språk som er valgt. Det er også mulig å sortere bokstavene eller ordene etter hvilket språk de opprinnelig er lagt inn i systemet som. Det må så vurderes hvor ressurskrevende en slik løsning vil være og hvor avansert sortering som bør benyttes for systemet.

Det er kanskje ønskelig å starte med en litt enkel måte å sortere på. Dersom vi lager en sorteringstabell for bokmål først, så kan vi utvide sorteringen senere, ved å innføre slike for det språket man vil sortere på. Dette er for så vidt en akseptabel måte å gjøre dette på i og med at utvider fra dette språket, og det er det språket de fleste benytter seg av. Men for mange vil dette virke forvirrende siden man er vandt med å se ord sortert avhengig av hvilket språk man benytter. Det vil derfor være å anbefale at sorteringsmetoder for alternative språk kommer på plass så raskt som overhode mulig.

I tillegg til en språktabell må vi ha en algoritme som benytter tabellen. Det må også stilles krav til denne algoritmen om at den skal være så effektiv som over hodet mulig. Det er også mulig å sortere på flere plasser i systemet, og til forskjellige tidspunkter. Vi må altså bestemme hvor og når sorteringen skal skje.

Som i de fleste andre systemer er det gjort en sortering før brukeren får en oversikt, men i tillegg kommer det en del forespørsler om å sortere på andre kriterier. På Altinn sin hovedside, får man alle skjemaene sortert på navn når man først kommer inn. Disse har faktisk en id og vil derfor sorteres likt uavhengig av språk, men man kan altså også velge å sortere etter "Sist endret av" og status, og kan dermed inneholde tegn man ikke bruker i bokmål.

Et kjent medium som har måttet løse dette problemet er telefonkatalogen. Her har de løst problemene med dobbel a og ö ved at de sorteres som det tilsvarende norske tegnet. Ö sorteres altså som en ø, mens aa sorteres som å. Böhmer og Böhmer sorteres med andre ord etter tilsvarende fornavn i telefonkatalogen, og det samme med Baardsen og Bårdsen. Dette er vel en måte å sortere på som de fleste som benytter norsk som hovedmål kan identifisere seg med, og som kan benyttes uten at man opplever noe større forvirring.

5.1.5 Oversetting

Det er to deler av Altinn som påvirkes av flerspråkligheten direkte, portalen og skjema eller dialogen hvor skjemaene kommer fra forskjellige leverandører, mens portalen er felles for alle disse. Selv om det er ønskelig å få en felles flerspråklig løsning for både skjema og portal, har disse litt ulike krav og må vurderes hver for seg.

Portalen:

- Begrenset antall felter
- Liten gjentakelse av felt
- Ingen fritekst som input

Det enkleste vil nok være å bruke feltløsningen for oversetting her. Mest sannsynlig er verdiene til feltene hardkodet på et eller annet vis. Det vil da være nødvendig å bytte disse tekstene med en verdi for hvert felt. Disse verdiene blir da brukt for å finne frem til riktig tekst på riktig språk ved hjelp av en enkel tabell. Ellers kan portalen stort sett være som den er i dag.

Skjema:

- Over 80 skjema
- Begrenset antall felter pr skjema
- Mye gjentakelse av felter på kryss av skjema
- Forskjellige leverandører/produsenter av skjema
- Noen felter tar fritekst som input
- Skjemaene skal kunne lastes ned hos utfyller, fylles ut der og leveres tilbake
- Alle skjema er ikke likt utformet

Totalt vil det være en god del felter i denne delen, og antallet øker fremdeles. I tillegg er det mye gjentakelse på kryss av skjema. Til tross for dette vil jeg tro at en feltløsning også her vil være det enkleste. Fritekstfeltene må da byttes ut med for eksempel med valglister. Det må med andre ord stilles litt strengere krav til skjemaene. Oversettingen mot kunden kan utvikles og skje i Altinn systemet, men selve tabellene for oversettingen må lages hos produsent av skjema.

Skjemaene er i dag definert etter XML Schema (XSD) standarden. Dette er et godt utgangspunkt for videre utvikling mot feltløsning av flerspråklighet. Det er enkelt å bytte tekstfelt i skjemaene med verdier som kan brukes for å hente ut teksten på riktig språk. Det som er hovedproblemet er altså å få en god måte å vise skjemaene på, eller med andre ord hente verdiene fra skjema, finne riktig tekst i tabellen, og vise dette til bruker/utfyller.

5.1.6 Andre vurderinger

Det er selvsagt en del andre ting å vurdere i et slikt stort system. Påvirker feltløsningen sikkerheten eller effektiviteten til systemet negativt? Er arbeidet med å gjøre om systemet så omfattende at det ikke lar seg gjøre med de ressursene vi har tilgjengelig, eventuelt er det noe annet som er viktigere for øyeblikket? Er det eksisterende teknikker som kan gjøre arbeidet for oss?

Siden Altinn er delt inn i 4 deler, der sikkerheten er en egen del, kan man sørge for at feltløsningen bare omfatter de andre delene av systemet. Det er heller ingen grunn til at sikkerhetsdelen som ikke vises for brukeren skal være flerspråklig. Det må altså mest sannsynlig menneskelige feil til for at feltløsningen skal påvirke sikkerheten.

Effektiviteten til systemet derimot vil bli påvirket til en viss grad. Siden det må hentes en ekstra tabell med en kolonne for hvert språk, blir det en ekstra belastning for systemene. Det må leses mer fra tabellen, det må gjøres litt mer arbeid i prosessering for å hente tekstene fra tabellen, og det tabellen må overføres over nettet.

Når man skal regne ut hvor stor innvirkning slike ting har på systemet ser man ofte bort fra prosesseringen. Dette er fordi lesing/skriving til disk og overføring over nett tar mye lengre tid enn prosessering. Også i feltløsningen vil prosesseringen utgjøre en så liten del, at man kan se bort fra den. Ofte er det også innsending av data til systemet, og lagring av denne som utgjør flaskehalsen på systemet. Siden systemet er skalert for å kunne takle siste liten innrapportering av for eksempel selvangivelse, og med tanke på at innsending av data, og lagring av data, krever mer ressurser enn lesing av data, kan man anta at det ikke blir noe problem.. Likevel er det viktig å effektivisere jobben så mye som mulig, slik at man i minst mulig grad gjør systemet mindre effektivt.

Kapittel 4.2 gir innblikk i hva som trengs for å lage et flerspråklig system. I tillegg må løsningen integreres inn i det eksisterende systemet, noe som ikke er diskutert i særlig grad.

Det finnes en del teknikker og systemer som kan hjelpe med å gjøre arbeidet, men ingen som kan gjøre alt arbeidet for oss. Nynorsk har allerede blitt nevnt og er et godt eksempel. Programmet forenkler arbeidet med å gjøre om fra bokmål til nynorsk. Det finnes mengder av algoritmer for sortering, noen av disse passer utmerket til sortering av tekster, og må bare tilpasses for å sortere etter sorteringstabellene.

5.2 Feltløsning for Altinn

Feltløsningen er en løsning som ser ut til å passe særdeles godt i Altinnsystemet. Kapittelet forsøker å vise med eksempler hvordan en feltløsning for Altinn kan fungere.

5.2.1 Språktabell

Tabell 5 viser hvordan de nordiske tegnene kan sorteres på alle nordiske språk. En del av tegnene finnes ikke på alle språkene i løsningen, men i det tilfellet vil de heller ikke ha behov for å sorteres. Et problem er ”dobbel A”, som på norsk skal sorteres som en ”Å” dersom den uttales på den måten. Den enkleste måten vil være å håndtere alle disse likt og som to A-er. Mer riktig vil det være å håndtere alle som å, noe denne tabellen viser en måte å gjøre det på. For å kunne vite når den skal være det ene og ikke det andre krever utvidede regler og kan ikke gjøres enkelt ved hjelp av en slik tabell.

Om man senere skulle ønske å utvide med svensk, tysk og til og med finsksvensk, så måtte man lage tilsvarende tabeller for disse språkene, for å få sorteringen etter språkvalg. V W problematikken kan løses ved å sette disse til å ha samme verdi, slik at man ser på neste bokstav, mens Ö på svensk settes opp i den sorteringstabellen til det tilhørende språket.

A	Á	B	C	Č	D	Ð	E	F	G	H	I	J	K	L	M	N	Ŋ	O	P
1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37	39
a	á	b	c	č	d	đ	e	f	g	h	i	j	k	l	m	n	ŋ	o	p
2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40

Q	R	S	Š	T	Ŧ	U	V	W	X	Y	Z	Ž	Æ	Ä	Ø	Ö	Å	Aa
41	43	45	47	49	51	53	55	57	59	61	63	65	67	69	71	73	75	75
q	r	s	š	t	ŧ	u	v	w	x	y	z	ž	æ	ä	ø	ö	å	aa
42	44	46	48	50	52	54	56	58	60	62	64	66	68	70	72	74	76	76

Tabell 7: Sorteringstabell for nordiske tegn, slik tegn vil sorteres på bokmål, nynorsk og samisk.

Den tyske ”Ö”-en er ikke fullt så enkel å plassere i denne tabellen. Siden den sorteres likt som to sammensatte tegn, oe, kan man ikke gi den en heltallsverdi slik at den blir plassert riktig.

Alle språk har som nevnt tidligere et alfabet med tilhørende sorteringsregler. Sorteringstabellene må altså knyttes til et språk. Tabell 6 er ikke representativ for sorteringsreglene for alle språkene ønskelig i systemet. Engelsk har ingen av de såkalte spesialtegnene, som æ, ø og å. Siden språket ikke har tegnene har man heller ingen regler for hvordan disse tegnene skal sorteres. Det er likevel nødvendig å ha regler for disse når

egennavn og lignende ofte har disse tegnene. Det er mulig å benytte samme sorteringstabell også for engelsk. Den eneste regelen som bryter med engelsk direkte er dobbel A. De resterende tegnene bør sorteres slik man antar det vil virke mest naturlig for det språket. Som eksempel kan man nevne ordet ”gründer” på norsk. Ordet brukes, men har et tegn som ikke finnes i det norske alfabetet. De fleste vil sortere ordet som ”grunder”, men det finnes ingen regler for dette direkte.

Sorteringstabellen for engelsk vil med andre ord kunne være lik tabell 6 med unntak av dobbel a. Likevel er det enkelte tegn som engelsktalende uten kjennskap til norsk vil stusse litt på når det gjelder sorteringen. Dette er spesielt de særnorske og svenske tegnene. En engelskmann vil kanskje se på disse som spesielle utgaver av A og O, og sortere etter dette. For eksempel Å rett etter A.

5.2.2 Sortering

Sortering av tekst er som nevnt tidligere ikke fullt så enkelt som annen sortering. I sortering av tall og andre størrelser som veldig enkelt kan sammenlignes så finnes det ofte faste oppsett for hvordan man bør sortere disse. Dette finnes også til en viss grad for tekster også, men her er det mange muligheter for å tilpasse sorteringen etter hvilke behov man har.

Man kan absolutt benytte en del av de samme mekanismene man har for sortering av tall til å sortere tekststrenger også. For å kunne gjøre det kreves det noen tilpassninger. I stedet for å bare implementere en algoritme som sorterer tall i riktig rekkefølge, kan man for tekster dele arbeidet inn i 3 metoder:

1. Metode som finner verdien av et tegn/kombinasjon av tegn fra en sorteringstabell.
2. Metode som sammenligner to (eller flere) ord ved å benytte metode 1.
3. Metode/algoritme som sorterer ord i riktig rekkefølge ved å benytte seg av metode 2.

Metode 3 vil ofte være vanlige sorteringsalgoritmer med små tilpassninger til å benytte metode 2 og sortering av tekster i stedet for tall og lignende. Vanlige algoritmer kan være for eksempel quicksort, instertsort, mergesort og så videre.

Valget av sorteringsalgoritme vil ofte være bestemt av effektiviteten til algoritmen. Dessverre finnes det ingen algoritme som er den mest effektive i alle tilfeller. Noen er bedre på delvis sorterte tabeller, mens andre er best når tabellene er virkelig store. Man må derfor vurdere hva som passer best for sitt system, for man bestemmer seg for hvilken algoritme man vil benytte.

Ved å se litt nærmere på de tre delene sorteringen består av, kan man lettere forstå hvordan sorteringen faktisk skjer i systemet.

Første del prøver altså å finne verdien av et tegn fra sorteringstabellen. Dette virker i utgangspunktet ganske enkelt, men det finnes noen få ting som vanskeliggjør prosessen. Noen tegn i enkelte språk kan ha samme verdi. For eksempel "V" og "W" i finsksvensk. I utgangspunktet ikke noe stort problem, men det betyr at man ikke bare kan gå ut i fra plasseringen av tegnene i en tabell. Man må altså også ha med et felt, verdi. I noen språk kan sammensetningen av tegn gjøre at tegnene skal sorteres i annen rekkefølge enn det man normalt ville gjort. Et eksempel på dette er norsk "AA". Denne bokstavkombinasjonen skal som oftest tolkes som "Å" på norsk, og ikke som to a-er etter hverandre.

Omtrent det motsatte problemet har vi jo også med tysk "Ö", som skal sorteres som "OE". Med andre ord et tegn som skal sorteres som to tegn. Dette tilfellet gir oss en del problemer med sorteringstabellen. Å velge en enkel verdi her, slik at tegnet blir plassert på riktig sted er ikke enkelt. Det finnes heldigvis løsninger også på dette. En enkel løsning kan være å lage en egen tabell for spesialregler som dette. Dobbel A kan også med fordel falle inn under disse reglene. Disse spesialreglene kan da oppfylles, eller sjekkes etter at resten av sorteringen er ferdig, eller man kan gjøre den samtidig. Dette kommer da inn under metode 2 av sorteringen.

```
/*METODE FOR Å FINNE VERDIEN AV ET TEGN/ SAMMENSATT TEGN*/
private int finnSorteringsVerdi(String inn_streng)throws NumberFormatException{
    int streng_plassering=0;
    int streng_verdi = 0;
    String finn_streng = inn_streng;
    while(streng_plassering<sort_table.length){
        String tabell_streng = sort_table[streng_plassering][0];
        if(tabell_streng.equals(finn_streng)){
            break;
        }else{
            if(tabell_streng.equals(finn_streng.substring(0,1))){
                break;
            }
        }
        streng_plassering++;
    }
}
streng_verdi = Integer.parseInt(sort_table[streng_plassering][1]);
return streng_verdi;
}
```

Programkode 1: Første del: Metode 1: finnSorteringsverdi(). Finner verdien av et tegn eller sammensatt tegn, og returnerer dets verdi.

Andre del av sorteringen er å sammenligne to, eller flere, ord ved å benytte seg av metode 1. Man går med andre ord gjennom ordene tegn for tegn, og bruker metode 1 på hvert av tegnene helt til man finner ut hvordan ordene skal sorteres. Som eksempel kan vi bruke navnene "Alf" og "Are". Metoden sender så de første tegnene i hvert ord til metode 1, og får tilbake verdien. Finner ut at tegnene har samme verdi og går dermed videre til

neste tegn. Gjør det samme med disse og finner at "l" sorteres før "r". "Alf" sorteres med andre ord før "Are".

Også denne metoden virker i utgangspunktet ganske enkel og oversiktlig, men som nevnt tidligere arver metoden problemene beskrevet over. Et eksempel kan være "Aalborg" og "Andenes". De fleste ville nok kjenne igjen den danske byen Aalborg her, og raskt bestemme at den skulle sorteres sammen med andre ord som begynner med Å. For datamaskinen er det dessverre ikke fullt så enkelt. Det står to a-er, og datamaskinen tolker korrekt dette som to a-er.

Som nevnt tidligere kan man lage ekstra sorteringstabeller for tilfeller som dobbel a, og tysk ö. Tegn som sorteres på slik måte at de ikke vil kunne vurderes samtidig som de andre tegnene. Om man vil lage et system som skal kunne benyttes på absolutt alle språk, uten å gjøre forandringer i software når man legger til nye språk er nok dette en egnet løsning. I Altinns og de fleste andre nordiske systemer kan man derimot gjøre ting litt enklere.

```
/*METODE FOR Å SAMMENLIGNE TO ORD.  
 * TAR TO ORD SOM INPUT, BRUKER finnSorteringsVerdi() FOR Å BEREGNE OM ET ORD  
KOMMER FØR DET ANDRE.  
 * RETURNERER TRUE OM ORD1 KOMMER FØR ORD2. FALSE HVIS DE ER LIKE ELLER  
ORD2 KOMMER FØR ORD1*/  
private boolean sammenlignOrd(String ord1, String ord2){  
    boolean ord1_forst=true;  
    int bokstavteller1=0;  
    int bokstavteller2=0;  
    while(bokstavteller1<ord1.length()&&bokstavteller2<ord2.length()){  
        String ord1_b = ord1 + " ";  
        String ord2_b = ord2 + " ";  
        String sort_sub_1 = ord1_b.substring(bokstavteller1,(bokstavteller1+2));  
        String sort_sub_2 = ord2_b.substring(bokstavteller2,(bokstavteller2+2));  
        int sort_sub_1_verdi= finnSorteringsVerdi(sort_sub_1);  
        int sort_sub_2_verdi= finnSorteringsVerdi(sort_sub_2);  
        if(sort_sub_1_verdi==sort_sub_2_verdi){  
            if(finnSorteringsVerdi(sort_sub_1.substring(0,1))!=sort_sub_1_verdi){  
                bokstavteller1++;  
            }  
            if(finnSorteringsVerdi(sort_sub_2.substring(0,1))!=sort_sub_2_verdi){  
                bokstavteller2++;  
            }  
        }  
        if(ord1.length()==bokstavteller1+1){  
            ord1_forst=true;  
            if(ord2.length()==bokstavteller2+1){  
                ord1_forst=false;  
            }  
        }  
        break;  
    }else if(ord2.length()==bokstavteller2+1){  
        ord1_forst=false;  
        break;  
    }else{  
    }  
}
```

```

}else if(sort_sub_1_verdi<sort_sub_2_verdi){
    ord1_forst=true;
    break;
}else{
    ord1_forst=false;
    break;
}
}
bokstavteller1++;
bokstavteller2++;
}
return ord1_forst;
}

```

Programkode 2. Andre del: Metode 2: sammenlignOrd(). Benytter seg av metode 1, finnSorteringsVerdi() for å beregne hvilket av ordene som skal sorteres først.

To ting bør gjøres for at man enklere skal kunne sjekke om man har dobbel a i ordene. Det første er å sende to tegn av gangen til metode 1, som finner verdiene av tegn. Dette er av den selvsagte grunn at man ikke kan sjekke om man har to a-er etter hverandre ved bare å se på et tegn. Det andre er å flytte AA frem i sorteringstabellen, slik at man sjekker dette før man sjekker om tegnet er en A. Dette avhenger selvsagt av hvordan man har implementert den første metoden, og må tilpasses etter det. Metoden vil også kunne fungere på flere sammensatte tegn.

Tredje del og ofte den mest kompliserte del er selve sorteringsalgoritmen. Fordelen med å inndele i tre deler, slik som det er gjort her, er at man her kan bruke kjente algoritmer uten store tilpassninger. Med noen få justeringer kan man kanskje til og med få til noe gjenbruk av sorteringsalgoritmer allerede implementert i systemet.

Sorteringsalgoritmen velges som regel med tanke på effektivitet og hvor ressurskrevende den er. Det finnes ikke en superalgoritme som er kjappere og mer effektiv enn alle andre i absolutt alle situasjoner. Man bør derfor se på hvordan algoritmen skal jobbe, og hva den skal sortere, før man velger.

I vårt tilfelle Altinn er det svært ofte at ting skal sorteres, men det er ikke alltid at de ordene som skal sorteres inneholder tegn som den eksisterende sorteringsalgoritmen ikke kan håndtere. Et spørsmål er altså om den nye sorteringen skal erstatte den gamle totalt, eller om de skal fungere side om side. En annen ting er at mengden ord som skal sorteres hver gang er varierende, men sjelden av en veldig størrelse. Dette betyr at hvilken algoritme vi velger i Metode 3 ikke får så veldig stor betydning, og at man kanskje heller burde se på effektiviseringen av de andre to delene. Uansett kan det være en ide å velge en algoritme som er effektiv på både små og store tabeller. Hvor sorteringen fysisk skal gjøres spiller også en rolle i valget. Dersom man ønsker å gjøre noe av sorteringen hos klienten, eller brukeren av systemet, kan det også være en fordel at man bruker så lite ressurser som mulig. På denne måten forhindrer man at ting går svært tregt bare fordi man har en svak maskin.

Quicksort algoritmen bruker lite ressurser. Er ikke den aller raskeste på små tabeller, men fungerer godt på både middels store og store tabeller. Algoritmen er også ganske enkel å implementere og modifisere. Algoritmen virker med andre ord ganske egnet til denne bruken.

```

/*QUICKSORT ALGORITME SOM BENYTTER SEG AV sammenlignOrd() METODEN OVER*/
void QuickSort(String a[], int lo0, int hi0) throws Exception
{
    int lo = lo0;
    int hi = hi0;
    String mid;
    if ( hi0 > lo0){
        mid = a[ ( lo0 + hi0 ) / 2 ];
        while( lo <= hi ){
            while( ( lo < hi0 ) && ( sammenlignOrd(a[lo],mid) ) )
                ++lo;
            while( ( hi > lo0 ) && ( sammenlignOrd(mid, a[hi] ) ) )
                --hi;
            if( lo <= hi ) {
                swap(a, lo, hi);
                ++lo;
                --hi;
            }
        }
        if( lo0 < hi )
            QuickSort( a, lo0, hi );

        if( lo < hi0 )
            QuickSort( a, lo, hi0 );
    }
}

private void swap(String a[], int i, int j){
    String T;
    T = a[i];
    a[i] = a[j];
    a[j] = T;
}

public void sort(String a[]) throws Exception{
    QuickSort(a, 0, a.length - 1);
}

```

Programkode 3. Quicksort algoritme som benytter seg av sammenlignOrd() i programkode 2. Metoden sort() er den som kalles på for å starte algoritmen.

En annen måte å gjøre sortering på er, som nevnt tidligere i oppgaven, å først sortere med de eksisterende sorteringsmekanismene. For eksempel å benytte databasens sorteringsalgoritmer før man får resultatet fra den. Deretter kan man bruke en egen algoritme for å sortere på tegn som databasens algoritmer ikke takler. For at denne metoden skal kunne fungere må man enten erstatte de tegnene databasen ikke takler med

kodesnutter, som for eksempel ”æ” for æ, eller som Unicode kodene for tegnene. Disse kodene må da gjenkjennes som et enkelt tegn i sorteringstabellen. Nettleserne eller annen software gjør så om kodene til de riktige tegnene for brukeren.

Fordeler:

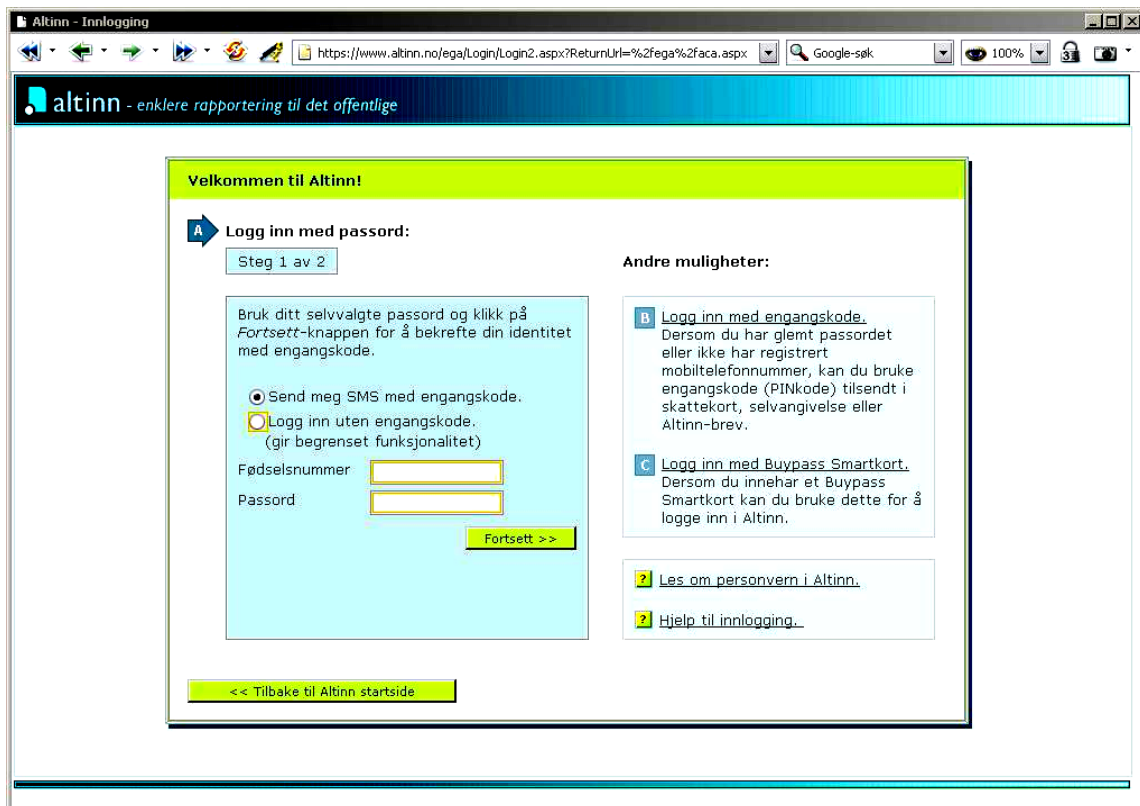
- Mange algoritmer er svært effektive på tabeller som allerede er noenlunde sorterte. De aller fleste algoritmer er bedre på noenlunde sorterte tabeller enn på de som ikke er sorterte fra før.
- Dersom det er unntaket heller enn regelen at man finner ord med tegn som ikke kan sorteres av databasesystemer, kan man slippe unna med en meget enkel egenprodusert sorteringsalgoritme. For eksempel en algoritme basert på insert sort.
- Man kan i større grad benytte systemet slik det er i dag. Alt man trenger å gjøre er å implementere en lite krevende sortering før bruker mottar resultatet.

Ulemper:

- Sortering er en veldig krevende prosess. Dersom databasesystemene skal stå for mesteparten av sorteringen, kan det gå ut over ytelsen. Eventuelt kan man bli nødt til å oppgradere hardware.
- Metoden kan ikke brukes dersom all sortering skal skje hos klient/bruker.

5.2.3 Oversetting

Selve oversettingen kan i og for seg også gjøres noenlunde enkelt. Bilde 2 viser hoveddelen av innloggingssiden i Altinn. Ser at vi kan dele inn i feltene som vist i tabell 6.



Figur 8: Logg inn hos Altinn. Bildet gjort skarpere for at det skulle bli mer lesbart i dokumentet.

Feltnummer	Tekst (Bokmål)
1	altinn – enklere rapportering til det offentlige
2	Velkommen til Altinn!
3	Logg inn med passord:
4	Steg 1 av 2
5*	Bruk ditt selvvalgte passord og klikk på Fortsett-knappen for å bekrefte din identitet med engangskode.
6	Send meg SMS med engangskode
7	Logg inn uten engangskode. (gir begrenset funksjonalitet)
8	Fødselsnummer
9	
10	Passord
11	
12*	Fortsett >>
13	Andre muligheter:
14	Logg inn med engangskode. Dersom du har glemt passordet eller ikke har registrert mobiltelefonnummer, kan du bruke engangskode (PINkode) tilsendt i skattekort, selvangivelse eller Altinnbrev.
15	Logg inn med Buypass Smartkort. Dersom du innehar et Buypass Smartkort kan du bruke dette for å logge deg inn i Altinn.
16	Les om personvern i Altinn.
17	Hjelp til innlogging.
18	<< Tilbake til Altinn startside
19*	Bruk ditt selvvalgte passord og klikk på Logg inn-knappen.
20*	Logg inn

Tabell 8: Felttabell av Altinn sin logg inn side. Studerer man siden, ser man at det blant annet er noen linker der. Dette betyr at en del av tekstene i tabellen skulle bestått av html-/javascriptkode i stedet for bare ren tekst. Dette er ikke tatt med av hensyn til lesbarheten av tabellen. Feltene 9 og 10 er feltene der fnr og passord skrives inn og står derfor tomme i utgangspunktet. Feltene merket med * vises avhengig av hva som er valgt i feltene 6 og 7.

Tabell 8 utgjør altså første del av feltløsningen. Den består av feltnummer og verdien som skal vises på bokmål. Andre steg vil da kanskje være å lage kolonner på de resterende språk. Med andre ord å oversette feltene til nynorsk, samisk og engelsk. Tabell 9 viser da hvordan kolonnen ser ut på samisk. Kolonnen med feltnummer er tatt med for lesbarhetens skyld. Når alle kolonnene er på plass er tabellen ferdig og kan benyttes av oversettelsesalgoritmen.

Feltnummer	Tekst (Samisk)
1	altinn – álkít raporterén almmolaš ásahusaide
2	Bures boahtin Altinnai!
3	Čálit sisa čoavddasániin:
4	1. oassi 2 oasis
5*	Ane iešválljejuvvon čoavddasáni ja deaddel Joatkke boalu duodaštan dihte iežat identitehta hávávalas čoavddasániin.
6	Sádde munnje hávávalas čoavddasáni SMS bokte
7	Čálit sisa hávávalas čoavddasáni haga. (ráddjejuvvon doaimmat)
8	Riegádannummar
9	
10	Čoavddasátni
11	
12*	Joatkke >>
13	Eará vejolašvuodat:
14	Čálit sisa hávávalas čoavddasániin. Jus leat vajálduhtán čoavddasáni dahje it leat registrerejuvvon mobiltelefonnummar, de sáhtát atnit hávávalas čoavddasáni mii čuožžu vearrokoarttas, iešdiedáhusas dahje Altinn reivves.
15	Čálit sisa Buypass Smartkortain. Jus dus lea Buypass Smartkort de sáhtát dan atnit go čálihat sisa Altinnai.
16	Loga personsuodjalusa birra Altinnas.
17	Sisačálihanveahkki.
18	<< Ruovttoluotta Altinna álgosiidui
19*	Ane iešválljejuvvon čoavddasáni ja deaddel Čálit sisa boalu.
20*	Čálit sisa >>

Tabell 9: Kolonnen for samisk. Oversatt av Risten Turi hos Samisk Språkråd. Turi gjorde oppmerksom på at oversettelsen ikke var korrekturlest, og at noen termer som ble lagt i forbindelse med oversettelsen ikke var kvalitetssikret.

Som nevnt tidligere må maskinen vite hvilket språk man vil vise siden på. Dette må første gang gjøres ved at brukeren selv trykker på en knapp eller link, velger språk fra en liste, eller på annen måte viser hvilket språk man vil ha. Neste gang man går inn på siden er det mulig for maskinen å huske hvilket språk man har ved for eksempel bruk av en såkalt cookie. Språkvalget bør gjøres så tidlig som mulig, men det er også en fordel om man kan bytte språk når man ønsker det. For eksempel så vil kanskje en same forsøke å bruke den samiske løsningen. Dersom han da kommer til et sted hvor han har problemer med å forstå, kan han da bytte over til bokmålsløsningen han er mer vant med. Om man legger språkvalget i rammen av portalen, blir den tilgjengelig uansett hva man holder på med i Altinn. Figur 8 prøver å vise hvordan dette kan se ut.



Figur 9: Header i Altinn med språkvalg

For å vise hvordan selve språkbyttet kan gjøres er det lagd et eksempelprogram i Java. Programmet tar utgangspunkt i "Logg inn" siden på Altinn. Programmet benytter seg av en lignende språktabell som tabell 8 og 9 viser kolonnene til, men med nynorsk og engelsk i tillegg. Kildekoden og språktabellen ligger i Vedlegg A. Det er brukt en del HTML koding for å lette formateringen i programmet, samt gjøre det hele mer leselig.

Figur 9 og 10 viser henholdsvis skjermbilder av bokmålsfeltene og de samiske feltene. Programmet starter opp med bokmålsfelter som standard. I et virkelig program, eller i den virkelige portalen vil det være gunstig om det er mulig å gjøre automatisk språkvalg når man har besøkt siden før. Siden dette er en applikasjon er det vanskelig å vise hvilke effektive metoder som finnes for å tilby denne funksjonaliteten på en webside.

altinn - enklere rapportering til det offentlige

Norsk(Bokmål) - NB

altinn - enklere rapportering til det offentlige

Velkommen til Altinn!

A Logg inn med pass...

Steg 1 av 2

Bruk ditt selvalgte passord og klikk på Fortsett-knappen for å bekrefte din identitet med engangskode.

Send meg SMS med engangsko...

Logg inn uten engangskode.(gir begrenset funksjonalitet)

Fødselsnummer

Passord

Fortsett >>

Andre muligheter:

B Logg inn med engangskode. Dersom du har glemt passordet eller ikke har registrert mobiltelefonnummer, kan du bruke engangskode (PINKode) tilsendt i skattekort, selvangivelse eller Altinn-brev.

C Logg inn med Buypass Smartkort. Dersom du innehar et Buypass Smartkort kan du bruke dette for å logge inn i Altinn.

Les om personvern i Altinn.

Hjelp til innlogging.

<< Tilbake til Altinn startside

Figur 10: Skjermbilde av eksempelprogrammet. Viser "Logg inn" siden på bokmål

Programmet har altså hardkodete verdier, som benyttes som standardverdier uten at noen språktabell trengs å leses. Når så et språkskifte blir kalt, ved å benytte comboboxen i øvre høyre hjørne, henter programmet ut valgt språk fra språktabellen. Det erstatter så verdiene i feltene med de verdiene som leses fra fil. I en komplett feltløsning ville feltene hatt en id, eller nummer i stedet for en standardverdi. Man ville da også ved oppstart måtte lese feltene fra tabellen. Siden dette programmet bruker metoder for å fylle opp feltene uten å benytte noen id annet enn feltnavn, kan det sies å være en halvveis feltløsning. Likevel er det et godt alternativ til programmer som allerede har hardkodete verdier i feltene.

altinn - álkít raporterén almmolaš ášahusaide Sami - SE

Bures boahtin Altinnai!

A Cálit sisa coavddasániin:
1. oassi 2 oasis
Ane iešválljejuvwon coavddasáni ja deaddel Joatkke boalu duodaštan dihte iežat identitehta háválás coavddasániin.

Sádde munnje háválás coavddasáni SMS bokte
 Cálit sisa háválás coavddasáni haga. (ráddjejuvwon doaimmat)

Riegádannumar
Coavddasátni

B Cálit sisa háválás coavddasániin. Jus leat vajálduhtán coavddasáni dahje it leat registrerejuvwon mobiltelefonnummar, de sáhtát atnit háválás coavddasáni mii cuožžu vearrokoarttas, iešdiedáhusas dahje Altinn reivves.

C Cálit sisa Buypass Smartkortain. Jus dus lea Buypass Smartkort de sáhtát dan atnit go cálihát sisa Altinnai.
Loga personsuodjalusa birra Altinnas.
Sisacálihánveahkki.

Figur 11: Skjerm bilde av eksempelprogrammet. Viser "Logg inn" siden på samisk

5.3 Beskrivelse av programvare

Det er utviklet to uavhengige programmer. Det første programmet , sorteringen viser hvordan man kan lage en sorteringstabell, og hvordan denne kan benyttes. Sorteringsalgoritmen som ble benyttet var en modifisert quick-sort algoritme, som benytter seg av metoden for å sammenligne ord.

Det ble raskt tydelig at å sortere på spesialreglene som dobbel a, og tysk ö var en virkelig utfordring, som ikke kunne løses direkte med en enkel tabell, som inneholdt tegnene i stigende rekkefølge. Det ble dermed nødvendig å lage egne regler for disse tegnene. Altinn skal ikke ha sortering på tysk i utgangspunktet, og det ble dermed, og av tidsmessige årsaker, besluttet å ikke vise i programmet hvordan dette kunne gjøres. Det er derimot gjort en drøfting av problemet, og vist der, hvordan det er mulig å gjennomføre.

Det andre programmet viser hvordan man kan implementere en særdeles enkel feltløsning i et eksisterende system, med små endringer. Adobe Form Server som Altinn benytter støtter ikke samiske tegn, og løsningen kan dermed ikke direkte implementeres inn i det eksisterende systemet. Programmet viser hvordan valg av språk kan gjennomføres, hvordan språkbyttet bør skje. Det viser også hvordan man kan hente ut felter i tabellen og vise disse på skjerm.

6 Kost/Nytte Analyse

Som nevnt tidligere er det vanskelig å beregne fortjenesten av å ha en flerspråklig løsning tilgjengelig. Vil likevel forsøke å beregne omtrentlig hva man kan forvente av direkte og indirekte fortjenester i et system som Altinn. Siden Altinn består av mange etater kan man gjøre analysen på hver og en av disse, eller man kan gjøre det for hele Altinn generelt.

Det man antar er at en god flerspråklig løsning vil kunne øke tilfredsheten med systemet, og kanskje øke bruken av det. Hos mange etater, for eksempel Skatteetaten får man ingen direkte fortjeneste av at kundene blir mer tilfredse. Alle må levere mva oppgaven, enten de liker måten det gjøres på eller ikke. Det som derimot kan utgjøre noe, er at ikke alle må levere elektronisk. Tabell 8 viser oversikten av kost/nytte analysen for etatsdialog i Altinn. Denne viser hva man tjener på at man leverer elektronisk i stedet for på papir. Denne kan man da også direkte overføre til denne oppgaven. For hver ny bruker som velger den elektroniske løsningen i stedet for papirløsningen får vi en fortjeneste på 16 NOK per transaksjon for etatene og 8 NOK for næringsliv. Potensialet i tabellen er hva Altinn hadde med de opprinnelige etatene. Altinn har vokst betraktelig siden da. Det vil si at potensialet har økt i takt med utvidelsen.

	Etat		Næringsliv	
	Altinn	Papir	Altinn	Papir
Innsending - porto/transaksjon	NOK 8,00	NOK 0,00	NOK 0,00	NOK 8,00
Behandling av mottak	NOK 7,00	NOK 28,00	NOK 0,00	NOK 0,00
Sending svar - porto/transaksjon	NOK 1,00	NOK 4,00	NOK 0,00	NOK 0,00
Kostnader	NOK 16,00	NOK 32,00	NOK 0,00	NOK 8,00
Innsparinger med Altinn	MNOK		MNOK	
Potensial	1000000	16	8	
<i>Realiseres av eborgere/ebedrifter som aksepterer ren elektronisk kommunikasjon som bokføres i Altinn dialogarkiv</i>				

Tabell 10: Etatsdialog i Altinn – kost/nytte.

Potensialet av den flerspråklige løsningen kan vel ikke sies å være det samme som potensialet av Altinn generelt. Størsteparten av brukerne til Altinn vil bruke bokmålsløsningen og bryr seg lite eller ingenting om det finnes alternative språk. Slik er det i hvert fall den dag i dag. Men ved å se på hvor mange som ikke er helt fornøyde med løsningen, eller som ikke har brukt løsningen kan vi finne ut litt mer om potensialet det flerspråklige løsningen har.

En undersøkelse ble gjennomført i desember 2004 av Perduco blant 2000 bedrifter for Altinn sentralforvaltning ved Brønnøysundregistrene. Undersøkelsen blant 2000 bedrifter, viser at 41 prosent av bedriftslederne i Norge kjenner til Altinn, mens 59 prosent ikke har hørt om Altinn. Blant de som har hørt om Altinn, oppgir hele to av tre at bedriften har rapportert og innlevert offentlige skjemaer gjennom Altinn. Undersøkelsen viser også at 58 prosent av de som har benyttet Altinn, mener at det er enkelt å bruke løsningen. 29 prosent er uenige og 13 prosent svarer vet ikke. Dette betyr at Altinn enda har mange potensielle brukere. Av de 29 prosentene som er uenige og 13 prosentene som ikke vet kan det kanskje være en del som synes løsningen hadde vært bedre om den hadde vært tilgjengelig på flere språk.

”Det er viktig både for bedriftene og samfunnet for øvrig at næringslivet bruker mindre tid på rapportering og mer tid på å skape verdier. Altinn er et hovedverktøy for å gi næringslivet en enklere hverdag. Derfor er det en prioritert oppgave å forbedre Altinn og sikre at stadig flere offentlige skjemaer legges inn i portalen”, sier nærings- og handelsminister Børge Brende. Ved å gjøre løsningen tilgjengelig på flere språk kommer man et steg nærmere denne oppgaven.

Å påstå at antallet brukere kommer til å øke betraktelig ved å kun innføre en flerspråklig løsning er vel å ta hardt i. Ifølge skatteetaten ba 138 personer i de samiske områdene om å få selvangivelsen på samisk. Siden selvangivelsen ikke finnes i elektronisk form på samisk vil disse måtte gjøre den for hånd. Man kan vel også anta at størsteparten av de som bruker nynorsk som hovedmål ikke vil bytte fra papirløsning til Altinn kun basert på at det foreligger en flerspråklig løsning. Det er nok mer oppfatningen av systemet som helhet som vil få personer til å velge det ene over det andre. Til tross for dette kan man anta at en flerspråklig løsning vil bedre helhetsinntrykket av systemet, og på den måten bidra til at man får flere brukere. Potensialet til en flerspråklig løsning vil dermed, sammen med resten av det som forbedrer helhetsinntrykket være alle innrapporteringer som ikke allerede blir gjort av systemet.

Som nevnt tidligere består Skatteetatens oppgaver blant annet å være i dialog med den som har rapportert inn for å rette opp feil og missforståelser som blir gjort. Skatteetaten skiller ikke mellom feil som blir gjort og annen kontakt med kunden, og har derfor heller ingen statistikk over dette. Likevel vil en flerspråklig løsning kunne begrense disse feilene og missforståelsene i noen grad. Dersom flere innrapporteringer kan gjøres automatisk uten at man må inn å rette opp slike ting, vil man få en direkte innsparing her.

Generelt sett er det vanskelig, ut i fra dette, å bedømme helt nøyaktig om de økonomiske fortjenestene vil kunne dekke kostnadene man har. Om man velger å se separat på

språkene, vil man nok finne at den samiske løsningen har svært få potensielle brukere, og vil dermed ha en lav inntjening. Den nynorske løsningen derimot har kanskje det største potensialet, men siden de fleste som har nynorsk som hovedmål sjelden har noe problem med å bruke bokmål, vil man kanskje ikke få mange nye brukere, men heller flytte en del over fra bokmålsversjonen. Med andre ord er det den engelske løsningen som vil stå for det meste av inntjeningen. Slik det er i dag, er kanskje potensielle engelsk talende brukere færre enn nynorskbrukere. Det som gjør at disse står for større del av inntjeningen, av den flerspråklige løsningen, er mange av disse ikke kan norsk. Mange bedrifter innen oljeindustrien og andre med stor andel engelsk, ville også kunne spare en del på å bruke den engelske løsningen. Utlendinger kunne rapportert via Altinn, og man ville åpnet opp for en del nye tjenester og en dialog med resten av den engelsktalende verden.

7 Andre systemer og løsninger

En rekke systemer har allerede implementert flerspråklighet. Andre har diskutert problemet tidligere. Kapittellet omhandler en del av disse.

7.1 Acharya

Et prosjekt i India som jobber med å utvikle flerspråklige brukergrensesnitt i et land som har 18 offisielle språk, og der de fleste ikke kan engelsk. Alle språkene er fonetisk baserte, men skrivemåten varierer. Mange av språkene har ord som er felles som igjen ser ut til å ha røtter i språk som sanskrit og tamil. Den fonetiske naturen av språket fører til et skriftspråk som representerer lyd gjennom unike symboler. Hvert språk har sin egen måte å beskrive lyden og dermed et eget tegnsett. Skrivemåten for de fleste indiske språk består av symboler for rundt seksten vokaler og så mange som 35 konsonanter. Stavelser som er dannet fra disse basislydene har også en unik representasjon. Et konjunkt er begrepet på en stavelse dannet av en eller flere konsonanter og en vokal. Selv om man teoretisk kan tenke seg til flere tusen konjunker, er bare ca 800 av disse i vanlig bruk. Hver av disse kan igjen kombineres med en vokal og lage nesten 13000 individuelle lyder, som igjen skal ha sin unike representasjon i skriftspråket.

Skriftspråket består interessant nok bare av rundt 200 symboler som former unike former, som representerer konjunktene ved å kombinere former. Dette skjer ved å legge til ligaturer. For hvert språk er det veldefinerte regler for hvordan man skal skrive konjunktene og hvordan de kan kombineres med vokaler. Akshara er et begrep man normalt bruker om enkle kombinasjoner av en vokal og en konsonant, mens Samyuktakshara er brukt når man snakker om konjunktene.

Å behandle indiske språk på en datamaskin er altså komplisert av at hver av aksharaene eller samyuktaksharaene skal kunne skilles fra hverandre. Selv om bare noen hundre primitive former kan være tilstede i praksis for å lage kombinasjonene, må antallet aksharaer identifiseres om man skal få til tekstprosessering og lingvistikk.

Metodene som fungerer godt for et begrenset sett av rundt 26 bokstaver i det romanske alfabetet, vil selvsagt ikke bli tilstrekkelig når de benyttes her. Ikke bare på grunn av at det finnes tusenvis av aksharas, men også at det er mer enn en akseptabel måte å skrive disse kombinasjonene. Selv om det er klare regler for hvordan disse kan kombineres, tillater disse reglene forskjellige representasjoner av samme konjunkt, selv i samme språk, men ikke minst på tvers av språkene.

Det var altså et behov for å se på hvordan man representerte store sett av aksharaer slik at man lage en standard for hvordan man kunne representere og prosessere indiske språk. Det har blitt foreslått en rekke måter å løse problemene på. Mange av problemene er de samme som vi hadde med samisk, bare med mange flere tegn.

7.1.1 Font basert data innsetting

Denne metoden baserer seg på en funksjon støttet av konvensjonelle ord prosessorer hvor fonten som blir brukt for å vise teksten, kan velges dynamisk mens man skriver inn data. I dagens operativsystemer er fontvisningen såpass avansert at den kan takle at formen av et tegn kan være bygd opp av flere primitive former. I fonter for det indiske alfabetet vil da disse tegnene inkludere matraer, som er spesielle samyuktaksharaer og spesielle tegn, i tillegg til basistegnene for vokaler og konsonanter. Når en font er valgt i et slikt system, vil altså fonten bygges opp av hvilke tastetrykk man gjør på et vanlig ”qwerty-tastatur”.

Metoden har dessverre en del mangler:

- Tastaturet har taster for å generere 96 tegn. Utover dette må man taste inn koder ved å holde ALT knappen pluss en tallkombinasjon for å frem de øvrige mulighetene. Som sagt er det behov for rundt 200 tegn. Dette betyr at metoden er ganske tungvindt.
- Kombinasjonene av tegn for å danne matraer og lignende er font spesifikk. Dette betyr at man har forskjellige måter å representere hvert forskjellig språk. Dette betyr igjen at man ikke på noen enkel måte kan bruke denne måten til å oversette mellom språk. Fontene følger ingen standard, siden det ikke eksisterer noen standard, og dessuten er det ikke sikkert det er mulig å lage dette.
- Siden skriftene er representert av tegn som ikke har noe med språkreglene å gjøre er det særdeles vanskelig å behandle tekstene på noe vis, som å sortering, stavekontroll osv.

Fordelen er at det er god støtte for metoden, og at det gjør det enkelt å klippe og lime fra tekstene i vanlige nettlesere og lignende.

7.1.2 Translitterasjon basert på data innsetting

Dette er i hovedsak en metode for å lage dokumenter som kan skrives ut med indiske skrifter. Ideen er at man bruker romanske bokstaver for å representere aksharaene i språkene og gjør om resultatstrengene ved hjelp av spesielle dataprogram for å produsere utskrifter. Et av de første programmene som implementerte dette var Dvng prosessor for Devanagari som produserte en TeX-fil som ble et typesett for å vise skriptene i TeX programvaren.

7.1.3 Manuell skrivemaskin innsetting

Også denne metoden har vært tilgjengelig en stund i indiske systemer. Disse tilbyr et minimalt sett av aksharaer som består av basiskonsonantene og vokalene sammen med matraene slik at tekst kan lages slik som den skrives på språket. Tekstene som kan skrives mangler en god del konjunker, men kan være tilstrekkelig for vanlig dagligdagse tekster.

7.1.4 Spesielle programmer for å lage websider

I det siste har det kommet en rekke programmer som kan lage websider på en rekke språk. Dette inkluderer også de indiske. Metoden består av at ascii tekst kan benyttes for å representere en rekke tegn, ved å lage koder for de enkelte tegnene. Først i web dokumentet angir man hvilket tegnsett man benytter. Dersom så browseren kjenner igjen tegnsettet, skal den klare å vise tegnkombinasjonene på riktig sett. For eksempel kan følgende kodesnutt skrives for å representere en enkel matra.

```
<html>  
<font face="sanskrit 1.2"> s&lt;Sk&laquo;tm! </font>  
</html>
```

Som da resulterer i følgende matra i en webleser som takler Sanskrit 1.2:

संस्कृतम्

7.1.5 Phonetisk kartlegging av vokaler og konsonanter

Enda en måte å se på data i indiske språk er å tenke at teksten består av aksharaer som alltid kan bli brutt ned til vokaler og konsonanter, og kanskje noen symboler. I denne fonetiske metoden er et tastetrykk assosiert med hver vokal og konsonant, mens et program lager aksharaene ut i fra dette. Metoden ligner en god del på translitterasjonsmetoden, men her tilegner man ikke vokalene og konsonantene til en eller flere bestemte taster.

7.2 Flerspråklige løsninger

Det finnes allerede en del systemer med flerspråklige løsninger. Mange av disse fungerer svært godt til sin bruk, mens andre fungerer mindre godt. I Norge er det slik at mange legger inn noe nynorsk og resten bokmål for å oppfylle språkloven, men noen har valgt andre løsninger.

7.2.1 Norge.no

”Internettportalen www.norge.no ble lansert 24. januar 2000, etter at Regjeringen og styret i Kommunenes Sentralforbund (KS) hadde besluttet at det skulle etableres en felles inngang til all offentlig informasjon på Internett. Tjenesten skal bidra til at forvaltningen

framstår som helhetlig. Både borgerne og forvaltningen er målgrupper for tjenesten.”
(Norge.no)

Norge.no tilbyr informasjon på nynorsk og engelsk, på henholdsvis www.noreg.no og www.norway.no. Bildene 4,5 og 6 viser henholdsvis norge.no, noreg.no og norway.no, som er de tre portalen på bokmål, nynorsk og engelsk. Et kjapt blikk på sidene viser at nynorskutgaven og bokmålsutgaven har samme innhold, og er også bygd opp på nøyaktig samme måte. Den engelske utgaven har derimot annet innhold, men følger samme stil som de andre sidene. Den engelske utgaven er rettet mot folk som ikke bor i Norge og trenger derfor heller ikke være konsistent med de andre løsningene.

Ved å studere linkene ser man også ganske raskt et mønster. Portalen forsøker konsekvent å skille linkene på samme måte som det er gjort på forsiden, ved å bytte ut ”norge” med ”noreg” eller ”norway”. Dette gjør det enklere å vedlikeholde siden, enklere å finne frem til samme side på de andre språkene, og enklere å linke sidene sammen. Siden den engelske utgaven ikke har samme innhold som de to andre, vil dette ikke gjelde i like stor grad her.

Det er imidlertid noe som kan virke litt underlig med den engelske løsningen. Om man for eksempel følger linken kalt ”Arbeidsmarked” fra bokmålsløsningen, får man opp en oversikt over lenker i forbindelse med dette temaet. Om man så ønsker å sende denne linken til en engelskbruker, og bytter ut ”norge” med ”norway” får man opp en litt merkelig løsning. Rammen i portalen er på engelsk, mens hovedvinduet er på nynorsk.

norge.no - Startsidene

http://www.norge.no/

Google-søk

100%

norge.no

...din veiviser i det offentlige

Tekstversjon | Om norge.no | Spør oss | English | Nynorsk

SØK

Finn myndighet
 Stortinget
 Sametinget
 Regjeringen
 Departementene
 Fylkesmennene
 Fylkeskommunene
 Kommunene
 Domstolene

Temasider A til Å
 Datoer og frister
 Offentlige skjema
 Kart
 Nyheter
 Nettstedskart

Norge.no
 Skrivarvegen 3
 6863 Leikanger

Grønt nummer:
 800 30 300
 E-post: info@norge.no

Samfunns-kartet

Datoer og frister

Kart

Kvalitets-merking

Finn myndighet

Nytt fra forvaltningen

Nye eneboliger billigere enn brukte i Oslo
 13.04.2005

Se opp for ikke-godkjente røykvarslere
 13.04.2005

Utslippene økte mer enn verdiskapningen
 13.04.2005

Søk om storviltjakt på nett
 13.04.2005

Høring - opplysningsplikt overfor Norges Bank
 13.04.2005

Flyktningers flyttemønster i Norge
 13.04.2005

Les flere nyheter...

Ti mest leste tema

- Ansettelse og arbeidsforhold
- Arbeidsmarked
- Bolig
- Familie, foreldre og barn
- Helsetjenester
- Reiseliv
- Skatt
- Trafikk og kjøretøy
- Trygd
- Utdanning

Alle tema fra A til Å...

Ofte stilte spørsmål

Hvem kan jeg kontakte for å få informasjon om etablering av egen bedrift?

Er det lov å brenne bråte?

Hvordan kan jeg finne ut hvilket likningskontor jeg hører til?

Flere spørsmål og svar...

Nytt fra Norge.no

Resultatet av ulike kvalitetstester spriker

Årsrapport for Norge.no 2004

Til kunnskapsbanken...

Finner du ikke fram?

Spør Norge.no!

Ring **800 30 300**
 Møt oss på [nettprat](#)
 E-post: info@norge.no
 Send SMS

Aktuelt nå

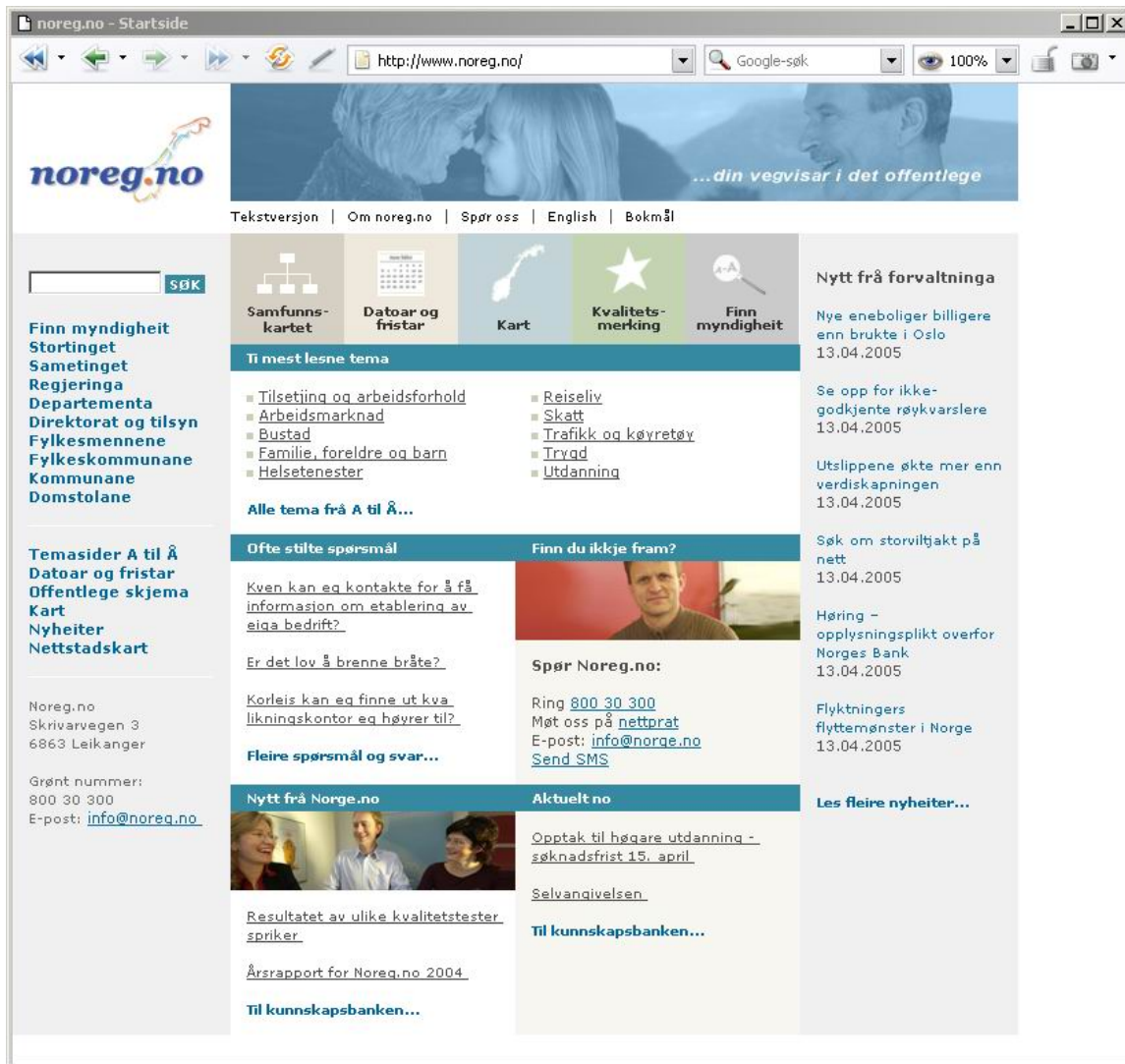
Opptak til høgere utdanning - søknadsfrist 15. april

Selvangivelsen

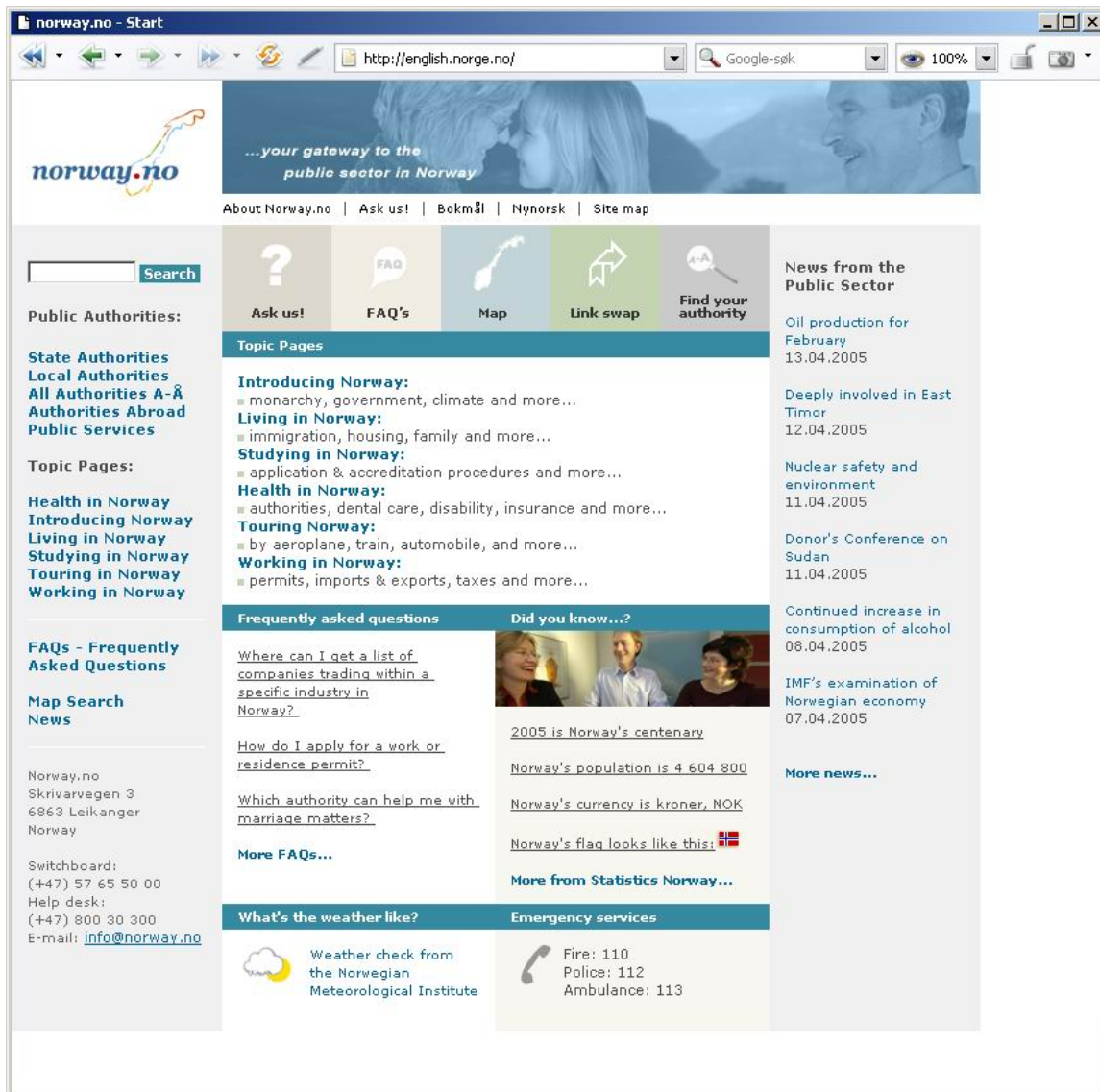
Til kunnskapsbanken...

Sist oppdatert: 14.04.05

Figur 12: Norge.no



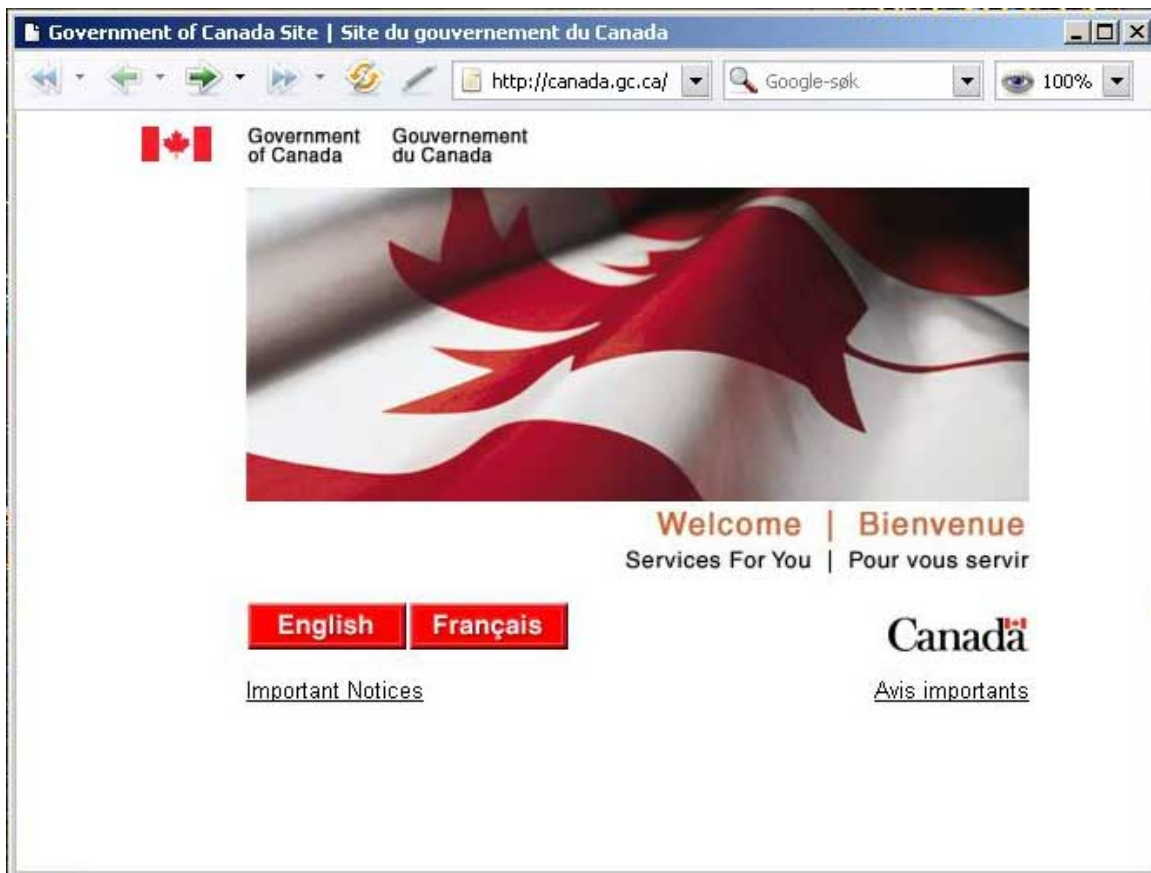
Figur 13: Noreg.no



Figur 14: Norway.no

7.2.2 Canada.ca

Canada.ca er en noe lignende portal for det kanadiske folk. Canada er et tospråklig land, der fransk og engelsk er helt likestilt. Bilde 4 viser forsiden til Canada.ca, som illustrerer hvor likestilt fransk og engelsk virkelig er i dette landet.



Figur 15: Forsiden til Canda.ca eller <http://canada.gc.ca>

På forsiden gjøres også valget for hvilket språk du vil ha videre inn i portalen. Språket vil være det samme så lenge du ikke velger å bytte. Språkvalget som gjøres på forsiden er en link til den engelske eller franske forsiden, men språkbytte kan skje uavhengig av hvor du er i portalen.

Når man bytter fra fransk til engelsk, inne i portalen, kan det virke som om man enten hopper til tilsvarende posisjon i den engelske portalen, eller at alle elementene skifter språk. Det siste kan tyde på at de har brukt en løsning som ligner på feltløsningen, men man ser også at adressene på sidene forandrer seg. Bilde 5 og 6 viser henholdsvis den engelske og den franske forsiden.



Figur 16: Den engelske forsiden til Canada.ca http://canada.gc.ca/main_e.htm



Figur 17: Den franske forsiden til Canada.ca. http://canada.gc.ca/main_f.html

Om man studerer adressefeltet på de forskjellige språkene, mens man gjør språkbyttet, kan man se at man har skilt naturlig mellom de franske og engelske sidene med å kalle dem for `_e` for engelsk eller `_f` for fransk. "`main_f.html`" er med andre ord den franske forsiden, mens "`main_e.html`" er den tilsvarende engelske. Disse skillene på adressene gjør det enklere å linke sammen de tilsvarende sidene på forskjellige språk. Dersom sidene enkelt og greit er linket sammen ved hjelp av html linker, vil det si at de har

*antall sider * antall språk * antall språk = antall linker.*

Metoden ser ut til å fungere meget bra, men man kan tenke seg til at det kan bli problemer om man måtte utvide med flere språk. Siden Canada er nøye med å

etterkomme kravene til urbefolkningen ville det kanskje være ønskelig å utvide portalen med noen av disse språkene også etter hvert. Dersom siden er bygd opp av en mengde linker, kan det bli temmelig mange linker å holde styr på med bare noen få ekstra språk. Dette vil jo da også føre til en temmelig stor jobb med å utvide flerspråkligheten.

7.2.3 Norge.no vs Canada.ca

Som vi skjønner er det mange likheter mellom disse portalene. Norge.no finnes også i nynorskutgaven noreg.no. De to versjonene er likestilte, slik som engelsk og fransk i Canada. Sidene virker tilnærmet identiske, uten at en av dem har noe fortrinn over den andre. Selv om Canada har litt strengere krav til flerspråkligheten, 50/50 imot den norske 30 % nynorsk regelen, er norge.no altså like god på nynorsk som på bokmål.

Språkbyttet virker i utgangspunktet likt. Inne i den canadiske portalen, ligger linken for å bytte språk godt synlig i øvre del av rammen, slik som i Norge.no. Mer enn dette er derimot ikke likt her. Der det canadiske språkbyttet bytter språk mens man fremdeles er på samme side i portalen, kastes man til forsiden ved språkbyttet i norge.no. Dette kan i beste fall virke irriterende og unødvendig på bruker. Om bokmålsbrukeren for eksempel vil vise en side til sin nynorske venn, kan han ikke bare bytte språk. Han må enten gjøre om på linken slik at det står ”noreg” i stedet for ”norge”, eller lete seg frem til der han var i portalen. Metodene fungerer naturligvis ikke for engelsk, siden denne har noe annet innhold og oppbygning. Det var mulig å finne sidene ved å bytte til ”norway”, men fikk da opp en blanding av engelsk ramme, og nynorsk innhold.

I Canada ser det ut til at man har lenket hver side til hverandre, slik at ”english” lenken i rammen fører til den tilsvarende engelske siden. Portalen fungerer utmerket og har ingen løsninger som gjør at den virker inkonsekvent eller medfører problemer for bruker. Portalen gjøres enda bedre ved at de fleste andre institusjoner og etater i Canada har bygd opp sidene etter samme modell. Dette gjør også at bruker blir godt vant med sidene, og vil ha små problemer med å benytte en annen side, som er bygd opp på samme måte.

Man kan påstå at den norske løsningen består av 3 portaler. For å lenke disse i hop har man lagd en link til forsiden av de andre portalene inn i rammen. Bokmålsportalen og nynorskportalen er bygd opp på samme måte, og skal også ha samme innhold. I tillegg har de lagd en fast oppbygning på lenkene. Det siste fører til at det blir enklere å tenke seg til hvor den tilsvarende siden på nynorsk befinner seg. Den Canadiske versjonen trenger ikke den siste mekanismen for å finne frem til tilsvarende dokumenter, men har altså valgt å gjøre det også. Dette har mest sannsynlig lettet arbeidet deres med å linke sider sammen, men tilbyr også ”enklere” tilgang til sider, ved å manipulere linker.

Norge.no har også forsøkt å opprettholde språkvalget når man forlater portalen, men siden de fleste nettstedene de linker til ikke har flerspråklige løsninger, er dette med begrenset suksess. De fleste sider har en informasjonsside på engelsk. Svært få har både bokmåls- og nynorskutgaver.

Løsningen til Canada.ca ligner bemerkelsesverdig på hvordan feltløsningen ville sett ut. Siden det er to språk med samme innhold, eller med andre ord to like deler med forskjellig språk, kan portalen med uten problemer ha benyttet feltløsningen i stedet for å linke sammen sidene. Som nevnt tidligere fungerer siden utmerket med bare to språk, men dersom den skal utvides, krever det mer arbeid enn det ville gjort med en feltløsning på plass.

Siden norway.no har annet innhold og oppbygging en de andre delene, kan portalen ikke like enkelt benytte feltløsningen. Det kunne derimot kanskje være ønskelig å ha informasjonssidene på flere språk enn bare engelsk. Som nevnt tidligere har er det mange i Europa som forstår spansk og fransk bedre enn engelsk. Størsteparten av turismen i Norge er av turister som ikke snakker engelsk som førstespråk. Dersom samme informasjon skal gis til spanske eller franske besøkende, som engelske, kan feltløsningen likevel komme til sin rett. Norge.no kan da deles inn i to portaler, en for nordmenn som prater norsk, og en for utlendinger. Begge portalene kan da dra nytte av feltløsningen. Den første med bokmål og nynorsk, eventuelt også samisk. Den andre med de resterende språk. Når feltløsningen er på plass er jo også muligheten for å utvide med ytterlige språk forenklet betraktelig.

7.2.4 Min Side

Kravspesifikasjonen til Min Side sier blant annet følgende om den kommende portalen:

”Det skal i løpet av 2005 utvikles en internettportal slik at innbyggerne kan møte det offentlige via en felles inngang. Dialogen med det offentlige skal bygges opp på innbyggernes premisser og ikke ut fra sektorenes og forvaltningens struktur og behov. (St. prp.nr.1, Moderniseringsdepartementet.)

Min side sin visjon er at den skal bidra til en enklere hverdag for folk flest ved å etablere en felles inngang til elektroniske tjenester, og til dialog med det offentlige. Min side skal dessuten øke muligheten for innsyn ved å gi samlet oversikt over informasjon som er registrert om den enkelte.

Min Side har tre målsettinger:

1. ”En enklere hverdag” for innbyggerne: Min Side skal være lett å finne fram i, tilby enkel pålogging til informasjon og tjenester og gi innsyn i data offentlig sektor har registrert om den enkelte innbygger.
2. Etablere fellesfunksjoner som tilbyr enkel pålogging og tilgang til offentlige tjenester: slik funksjonalitet kan bidra til å effektivisere tjenesteytingen som etatene tilbyr.
3. Bidra til å synliggjøre og konkretisere behovet for en felles offentlig arkitektur for elektronisk samhandling.

Min Side som skal markedsføres overfor innbyggerne som en del av www.norge.no. Grunnen er at Min Side ikke skal bli et enda et nytt frittstående tilbud som kjemper om

oppmerksomheten med alle andre tilbud fra det offentlige. Ved å gi Min Side en plass på hovedsiden til Norge.no, får brukeren tilgang til to typer informasjon og tjenester samme sted:

- Et tilbud om åpen informasjon, som tilrettelegges av redaksjonen i Norge.no
- En personorientert Min Side som krever pålogging og som inneholder ”Mine data” og tilgang til transaksjonstjenester

Målgruppen til Min Side er stor: den omfatter alle unge voksne og voksne i Norge. Den er dermed så sammensatt som befolkningen selv – og oppgaven med å tilpasse praktisk løsning og tjenestetilbud slik at den oppfattes god og relevant for alle er en betydelig utfordring. Så langt råd er, må løsningen fungere for brukere med store ulikheter med hensyn til motivasjon, utstyr og kunnskaper.

Andre krav, som muligheten til å velge mellom språk og mellom offisielle språkformer i Norge, kan påvirke designet samt bruken av og valg av tegnsett. Min Side skal f.eks. tilrettelegges for bokmål, nynorsk, fremmedspråk og samisk. Samisk krever støtte for Unicode og Min Side må derfor legge til rette for bruk av Unicode. ”

Min side blir altså på en måte liggende under norge.no og skal følge samme design. Portalen legger også til rette for flerspråklighet ved å benytte Unicode, men har ikke spesifisert i kravspesifikasjonen hvordan det skal løses ut over dette, noe som er et tegn på at noen flerspråklighet ut over det vi ser hos norge.no i dag, ikke vil komme med første versjon.

Brukere med alle språk skal ha tilgang på samme funksjonalitet. Systemet kan derfor bygges opp på omtrent samme måte som Canada.ca, men siden det blir lagt opp til en del flere språk enn i Canada kan feltløsningen være bedre egnet her.

7.2.5 Skolelinux

Skolelinux er en helhetlig dataløsning skreddersydd etter skolens behov og ressurser. Et ferdig datasystem for skolene, der skolen selv ikke trenger sette sammen enkeltkomponentene. Skolelinux er en Linux-distribusjon som bygger på Debian Linux. Målet er å lage en komplett løsning som er enkel å installere, drifte og vedlikeholde for skolene, med programvare på elevenes eget morsmål.

Skolelinux kommer med en standard arktitektur, og inneholder mange ferdigkonfigurerte tjenester.

- Sentral brukerkatalog gjør at man kan logge seg på med samme brukernavn og passord over hele nettverket
- Sentral filserver tilbyr tilgang på samme filer og innstillinger.

- Tynnklientløsning. Programmer kan kjøres på en sentral tjener, mens brukergrensesnittet vises på tynnklientene. Gir også enklere vedlikehold, siden hovedbyrden av vedlikeholdet er på tjeneren.
- Skriverdeling gjør skriveren tilgjengelig over alt i nettet.
- Proxyserver mellomlager filer fra nettet og gjør surfing raskere.
- I tillegg tilbys tjenester som nettsjener, e-postsystem og database.

Skolelinux, Debian og all tilhørende programvare er fri programvare. Det vil si programvare hvor brukere har full tilgang til kildekoden og kan gjøre hva man vil med den, så lenge man ikke fratår andre den samme retten. Dette gir brukere alle rettigheter, og full kontroll over programvaren.

Systemet baseres på nynorsk, bokmål og nordsamisk. Alle program som følger med distribusjonen skal være tilgjengelig på bokmål og nynorsk. De viktigste programmene vil etter hvert komme også på nordsamisk. For samisk vil de også tilrettelegge for samiske bokstaver (kodet etter Unicode-standard) og en sorteringsfunksjon som følger alfabetene i de samiske skriftspråkene.

7.2.6 Andre eksempler

Et annet eksempel på portaler med flerspråklighet er suomi.fi. Portalen virker særdeles godt gjennomtenkt og har finsk, svensk og engelsk løsning. Løsningen er noe lik løsningen i Canada. Språkbyttet kan skje når som helst uten at man havner på forsiden av portalen. Den virker også konsistent på alle tre språk, med samme innhold og oppbygning. Når man forlater portalen er språkvalget opprettholdt. Adressene er bygd opp med at man skiller med en /soumi/, /svenska/ eller /english/ slik at man om ønskelig kan benytte denne metoden å gjøre språkvalget. Med andre ord en løsning ganske lik den canadiske, men med tre språk.

Statistisk Sentralbyrå (ssb.no) ser ut til å ha valgt den på mange måter tradisjonelle måten å oppfylle språklova på. Her er deler av informasjonen på nynorsk, og andre deler på bokmål, men ingen mulighet for å velge det ene eller det andre. Derimot så kan man velge en del andre språk. Engelsk, tysk, fransk, spansk, portugisisk og russisk er alternativene til norsk, men kun den engelske delen har samme innhold som den norske. De resterende er kun informasjonssider. Språkskiftet her er gjort på samme måten som i Canada og finland, ved at man fortsetter der man var, men med norsk i stedet for engelsk, og vise versa.

7.3 IGLO prosjektet

I Europa snakkes det som nevnt tidligere mange språk. Dette kan gjøre det vanskelig for informasjonsflyt og utveksling av ideer. For å gjøre dette enklere har det tradisjonelt blitt

brukt et såkalt Lingua Franca, et språk som de fleste i regionen skjønner. Engelsk har mer eller mindre fått denne statusen i Europa, selv om det er svært mange som tross alt ikke snakker engelsk. I stedet for å fremme et enkelt internasjonalt språk, kan det være en ide å satse på flerspråklighet.

IGLO prosjektets ide var altså todelt, men samtidig en enkel ide. Siden det er ganske ressurskrevende å lære nye språk, kan man ved å skaffe seg en delvis kompetanse i språket gjøre seg tilstrekkelig forstått for mange situasjoner. Gjensidig forståelse innebærer at man forstår hverandre selv om man bruker forskjellige språk, som for eksempel i Skandinavia. Her kommuniserer nordmenn, svensker og dansker med hverandre på sine egne språk. Ideen er da at ved undervisning rettet mot delvis kompetanse, kan man øke mulighetene for gjensidig forståelse ved nært beslektede språk.

Prosjektet involverte syv germanske språk, og besto av syv institusjoner fra syv land. Universitetet i Tromsø var koordinerende institusjon, og ansvarlig for arbeidet med engelsk og norsk. Hagen University var ansvarlig for implementasjonen. De resterende institusjonene var ansvarlige for arbeidet med de andre språkene og besto av Salzburg, Antwerpen, Lund, København og Reykjavik.

8 Oppsummering og konklusjon

8.1 Generelle tilrådninger

Man møter mange utfordringer når man skal utvikle et godt flerspråklig system. Tegnssett, sortering og grammatikk bestemmes av språk, og kan ha stor innvirkning på hvordan systemene må utvikles.

Tegnssett

Unicode er det klart beste alternativet når man skal utvikle flerspråklige systemer. Tegnsettet støtter flere tegn enn noe annet sett, og har god støtte i det meste av programvare i dag. Ulempen er at en del gammel programvare ikke støtter tegnsettet, og heller ikke har muligheter for å implementere støtte. I tillegg kreves det installasjon av fonter i en del av den nyere programvaren for å kunne vise tegnene.

Sortering

Sortering er en av de mest kompliserte oppgavene man står ovenfor. Med nye tegn, trenger man nye måter å sortere på. Hvordan tegn skal sorteres, bestemmes av hvilket språk man benytter. Det er derfor en fordel om man kan benytte egne sorteringsregler for hvert enkelt språk. Siden sortering kan være kompliserte algoritmer, og dermed koste mye ressurser, kan det vurderes å lette på dette kravet. Dersom systemet for eksempel inneholder lite sortering på tegn som sorteres ulikt på forskjellige språk, kan man vurdere det til at det ikke er kostnadseffektivt å implementere språkbestemt sortering.

Automatisk oversetting

Automatisk oversetting er forsøkt forskjellig grad av suksess. Et program som utfører tilnærmet automatisk oversetting er Nyno. Programmet suksess avhenger mye av at bokmål og nynorsk ikke er to ulike språk, men heller forskjellige presentasjonsformer av samme språk. Likevel vil mekanismene, brukt i programmet, til en viss grad kunne benyttes på andre like språk, som for eksempel norsk og dansk.

Fritekst

En fullstendig automatisk oversettelse av fritekst fra et språk til et annet språk med ulik oppbygging er tilnærmet umulig. Å lage et program, med kjennskap til alle ords betydninger, språkenes grammatikk, ordtak og så videre vil ikke være mulig. Novellen som slutter med ”og mannen ble så sint at han slo på TV-en” vil ha forskjellige oversettelser alt etter hvordan programmet tolker setningen, og selv om det er mulig å legge inn betydningen av setningen, vil man ha problemer dersom det faktisk var ment å være tvetydig.

Feltløsning – kodbar informasjon

Oversettelse av kodbar informasjon er et noe enklere tilfelle. Ordbokmetoden og feltløsningen ble beskrevet i oppgaven, der feltløsningen var en forenkling av ordboken.

Løsningen består av at dokument deles inn i felt med identifikatorer. Det lages så en felttabell med verdien av feltene på alle språk.

Feltløsning - språkbytte

Canada.ca er nevnt i oppgaven som et system som har alle de estetiske sidene av en god flerspråklig løsning. Språkbyttet er elegant gjennomført ved at man fortsatt er på samme plass i portalen etter byttet. Det samme er mulig med feltløsningen. Språkbyttet i feltløsningen, vil innebære at feltene endrer språk, ikke at man henter en ny side.

Feltløsning - linker

Canada.ca består av en mengde linker som binder portalen sammen på fransk og engelsk. Dersom portalen skulle utvides med flere språk, vil få et meget stort antall linker å holde styr på. Dette er fullt mulig med få språk, men siden antall linker øker eksponentielt med antall språk, har den sin begrensning. Feltløsningen er noe enklere på dette feltet. I stedet for å måtte legge til nye linker i hvert eneste dokument/side i portalen, legger man bare inn en ny kolonne i felttabellen.

Feltløsning - Norge.no

Norge.no har også en noe lik løsning. Også her benyttes html for å linke sammen sidene. Linken fører her til forsiden av portalen. Dette gjør at hver gang man bytter språk, blir man kastet til forsiden. Det får det til å virke som det er tre forskjellige portaler, med en enkel link til de andre portalene. Løsningen gjør det vanskeligere å finne frem i de andre språkene, vanskeligere å henvise ting i portalen til brukere av andre språk, og kan virke forvirrende på bruker. Fordelen er at linkene er logisk oppbygd slik at man kan benytte disse til henvisning med små modifiseringer. Ved å benytte feltløsningen her kunne man som nevnt tidligere sluppet dette problemet, men siden den engelske delen er ment å være ulik de norske, kan man ikke benytte løsningen her.

Feltløsning - Bruksområder

Feltløsningen er egnet i alle systemer som skal ha lik informasjon på alle språk. Portaler, som for eksempel ssb.no og norge.no, der informasjonen skal være ulik på engelsk og på norsk kan altså ikke benytte feltløsningen på disse språkene. Det er derimot ingenting i veien for at man deler systemet inn i to deler, der en felttabell består av bokmål, nynorsk og samisk, og en tabell består av de resterende språkene.

Nytte

En av grunnene til at flerspråklighet ofte blir nedprioritert i større systemer er at man ikke ser noen økonomisk gevinst i å gjennomføre dette. I offentlige systemer i Norge kan de fleste benytte en løsning som kun eksisterer på bokmål. Det er derfor få økonomiske argumenter for en flerspråklig løsning. Likevel er det mange andre argumenter som bør tas i betraktning. Løsningen fører til bedre tilgjengelighet for enkelte, og bedre tilfredshet blant mange siden man kan benytte løsningen på sitt eget språk.

Feltløsningen tilbyr lik presentasjon av informasjon på flere språk, uten en større implementering bak. Feltløsningen bruker kun tilgjengelig programvare. Det gjør den til en billig, og realiserbar løsning for de fleste systemer i dag.

8.2 Tilråding for Altinn

Dette arbeidet gir følgende tilråding for Altinn.

Mål

I målene for Altinn står det at det skal være en flerspråklig løsning, med bokmål, nynorsk, samisk og engelsk. Portalen skal tilby samme informasjon til bokmålsbrukere som til de andre brukerne.

Feltløsning

Altinn kan på en enkel måte inndeles i felt, som i eksempelprogrammet tidligere i oppgaven, og systemet består av et begrenset antall felt. Alt dette tilsier at feltløsningen er et godt alternativ når flerspråkighet skal implementeres. Feltløsningen tilbyr like sider på alle språk, med enkelt språkbytte, slik at man kan benytte et språk, for så å bytte til et annet uten å havne på et annet sted i portalen. Feltløsningen er enklere enn løsningen til Canada.ca, fordi man ikke har behov for å linke alle språkene sammen. Det er heller ikke behov for å gjøre endringer i portalen utover å lese en annen kolonne i felttabellen når språkbyttet blir gjort.

Unicode

I tillegg til å takle de språkene som ble spesifisert, vil Unicode åpne opp for å kunne utvide med enda flere språk i fremtiden. Siden et av argumentene for feltløsningen er at systemet skal være lett å utvide, vil Unicode være et mye bedre alternativ som tegnsatt enn for eksempel ISO 8859-10. Unicode er også en anerkjent standard i resten av verden og har unike tegn, i motsetning til ISO 8859 som blant annet har flere utgaver av samme sett. Altinn har tidligere også påpekt at Unicode var nødvendig for å kunne gjennomføre et flerspråklig system i den grad de ønsker det. Altinn benytter Adobe Form Server som i denne utgaven ikke støtter samiske tegn. Det er altså ikke mulig med dagens system å oppfylle kravene. Det er derimot nevnt at dette skal være på plass i neste utgave av programmet, som er planlagt ferdig sommeren 2006.

Sortering

Sortering er som nevnt tidligere en utfordring i flerspråklige systemer. Det er en avveining mellom effektivitet og brukervennlighet for å finne ut hvilke sorteringsmekanismer som skal brukes. I Altinn er sortering et lite aktuelt problem. Skjema har nummer, og vil dermed ikke by på problemer når det gjelder sortering. Det man ellers kan sortere på er dato, også nummerisk, ”status” og ”sist endret av”. De to siste kan inneholde sære samiske tegn, men det er fullt mulig å omgå dette.

Inn-/utdata

Oppgaven har i stor grad dreid seg om å vise data på skjerm. For Altinn, og mange andre systemer, er det også en utfordring å motta data i en flerspråklig løsning. En engelsk bruker vil ha begrenset nytte av løsningen dersom han må levere data på norsk. Det er derimot ikke hensiktsmessig i Altinn å motta input på mange forskjellige språk. Det er derfor nødvendig å gjøre om på systemet slik at fritekst utenom numeriske verdier ikke sendes inn. (Latinske tall er felles for alle språk) Valglistene og valgbokser er en god

løsning på dette. Dersom man kan lage systemet slik at all input gjøres via slike valglister, og felter med numeriske verdier, vil ikke språk ha noe innvirkning på innrapporteringen. En slik løsning åpner også opp for mer automatisering, og vil kunne effektivisere behandlingen av en rekke dokumenter. Altinn er allerede i ferd med å gjennomføre noe som kan ligne på dette med en mer dialogbasert kontakt med brukeren, i stedet for papirbaserte skjemaer.

Oppgaveregisteret

Skjema i Altinn er skrevet av eksterne etater. Oppgaveregisteret bestemmer oppbyggingen av skjema. Det er derfor nødvendig at Oppgaveregisteret stiller strengere krav til leverandør av skjema. Nye modeller for skjema må lages. Disse må inneholde feltidentifikatorer og felttabell dersom skjemaet skal kunne være flerspråklig. Også her må det spesifiseres bruken av Unicode for at samiske tegn skal kunne benyttes. Spesifiseringene må også omhandle endringene for å gjennomføre automatiseringen. Fritekstfelter må erstattes med valglister, sjekkbokser og lignende.

Vedlegg A: Kildekode

Kildekoden til eksempelprogrammene i kapittel 5. Kodene viser programmer for sortering og feltløsning for logg-inn siden til Altinn. Selve felttabellen er også vedlagt.

Sortering

```
package language;
import java.io.*;
import java.util.*;

public class SorterTekst {
    String[][] sort_table;          //table of chars in order to be sorted by

    public SorterTekst()
    {
    }

    public void setSortTable(String[][] inn_sort_table){ //change order to sort by
        sort_table = inn_sort_table;
    }

    /*METODE FOR Å FINNE VERDIEN AV ET TEGN/ SAMMENSATT TEGN*/
    private int finnSorteringsVerdi(String inn_streng)throws NumberFormatException{
        int streng_plassering=0;
        int streng_verdi = 0;
        String finn_streng = inn_streng;
        while(streng_plassering<sort_table.length){
            String tabell_streng = sort_table[streng_plassering][0];
            if(tabell_streng.equals(finn_streng)){
                break;
            }else{
                if(tabell_streng.equals(finn_streng.substring(0,1))){
                    break;
                }
                streng_plassering++;
            }
        }
        streng_verdi = Integer.parseInt(sort_table[streng_plassering][1]);
        return streng_verdi;
    }

    /*METODE FOR Å SAMMENLIGNE TO ORD.
    * TAR TO ORD SOM INPUT, BRUKER finnSorteringsVerdi() FOR Å BEREGNE OM ET ORD
    KOMMER FØR DET ANDRE.
    * RETURNERER TRUE OM ORD1 KOMMER FØR ORD2. FALSE HVIS DE ER LIKE ELLER
    ORD2 KOMMER FØR ORD1*/
```

```

private boolean sammenlignOrd(String ord1, String ord2){
    boolean ord1_forst=true;
    int bokstavteller1=0;
    int bokstavteller2=0;
    while(bokstavteller1<ord1.length()&&bokstavteller2<ord2.length()){
        String ord1_b = ord1 + " ";
        String ord2_b = ord2 + " ";
        String sort_sub_1 = ord1_b.substring(bokstavteller1,(bokstavteller1+2));
        String sort_sub_2 = ord2_b.substring(bokstavteller2,(bokstavteller2+2));
        int sort_sub_1_verdi= finnSorteringsVerdi(sort_sub_1);
        int sort_sub_2_verdi= finnSorteringsVerdi(sort_sub_2);
        if(sort_sub_1_verdi==sort_sub_2_verdi){
            if(finnSorteringsVerdi(sort_sub_1.substring(0,1))!=sort_sub_1_verdi){
                bokstavteller1++;
            }
            if(finnSorteringsVerdi(sort_sub_2.substring(0,1))!=sort_sub_2_verdi){
                bokstavteller2++;
            }
        }
        if(ord1.length()==bokstavteller1+1){
            ord1_forst=true;
            if(ord2.length()==bokstavteller2+1){
                ord1_forst=false;
            }
            break;
        }else if(ord2.length()==bokstavteller2+1){
            ord1_forst=false;
            break;
        }else{
        }
        }else if(sort_sub_1_verdi<sort_sub_2_verdi){
            ord1_forst=true;
            break;
        }else{
            ord1_forst=false;
            break;
        }
        bokstavteller1++;
        bokstavteller2++;
    }
    return ord1_forst;
}

/*QUICKSORT ALGORITME SOM BENYTTET SEG AV sammenlignOrd() METODEN OVER*/
void QuickSort(String a[], int lo0, int hi0) throws Exception
{
    int lo = lo0;
    int hi = hi0;
    String mid;
    if ( hi0 > lo0){
        mid = a[ ( lo0 + hi0 ) / 2 ];
        while( lo <= hi ){
            while( ( lo < hi0 ) && ( sammenlignOrd(a[lo],mid) ) )
                ++lo;
            while( ( hi > lo0 ) && ( sammenlignOrd(mid, a[hi]) ) )
                --hi;
            if( lo <= hi ) {

```

```

        swap(a, lo, hi);
        ++lo;
        --hi;
    }
}
if( lo0 < hi )
    QuickSort( a, lo0, hi );

if( lo < hi0 )
    QuickSort( a, lo, hi0 );
}
}

private void swap(String a[], int i, int j){
    String T;
    T = a[i];
    a[i] = a[j];
    a[j] = T;
}

public void sort(String a[]) throws Exception{
    QuickSort(a, 0, a.length - 1);
}
}
}

```

Programkode 4. Klassen SorterTekst som utfører sortering av tekster på en angitt sorteringstabell

```

package language;
import java.sql.Timestamp;
import java.util.*;
import javax.swing.JFrame;

class testSort
{
    /**
     *
     * @param args
     */
    public static void main(String[] args)
    {
        SorterTekst st = new SorterTekst();
        //String[][] sort = {{"a","1"}, {"b","2"}, {"c","3"}};
        String[][] sort = {{"Aa","75"}, {"aa","76"}, {"
", "0"}, {"A","1"}, {"a","2"}, {"Á","3"}, {"á","4"}, {"B","5"}, {"b","6"},
{"C","7"}, {"c","8"}, {"Č","9"}, {"č","10"}, {"D","11"}, {"d","12"}, {"Ď","13"}, {"ď","14"}, {"E","15"},
{"e","16"}, {"F","17"}, {"f","18"}, {"G","19"}, {"g","20"}, {"H","21"}, {"h","22"}, {"I","23"}, {"i","24"},
{"J","25"}, {"j","26"}, {"K","27"}, {"k","28"}, {"L","29"}, {"l","30"}, {"M","31"}, {"m","32"}, {"N","33"},
{"n","34"}, {"D","35"}, {"ŋ","36"}, {"O","37"}, {"o","38"}, {"P","39"}, {"p","40"}, {"Q","41"}, {"q","42"},
{"R","43"}, {"r","44"}, {"S","45"}, {"s","46"}, {"Š","47"}, {"š","48"}, {"T","49"}, {"t","50"}, {"F","51"},
{"t","52"}, {"U","53"}, {"u","54"}, {"V","55"}, {"v","56"}, {"W","57"}, {"w","58"}, {"X","59"}, {"x","60"},
{"Y","61"}, {"y","62"}, {"Z","63"}, {"z","64"}, {"Ž","65"}, {"ž","66"}, {"Æ","67"}, {"æ","68"}, {"Å","69"},
{"å","70"}, {"Ø","71"}, {"ø","72"}, {"Ö","73"}, {"ö","74"}, {"Ä","75"}, {"ä","76"}};
        st.setSortTable(sort);
    }
}

```

```

String tabell[] =
{"Østersund", "Østfold", "Aalborg", "Åre", "Čálit", "Cannes", "Alstahaug", "Östersund", "Canberra", "Čun
ovuohppi fjellstue", "små", "Store", "n"};

try{
st.sort(tabell);
}catch(Exception ex){
System.out.println("ERROR"+ex);
}
for(int i=0;i<tabell.length;i++){
System.out.println(" "+ tabell[i]);
}
}
}
}

```

Programkode 5. Testmetode på klassen SorterTekst

Feltløsning

```

package language;
import java.awt.*;
import java.io.*;
import java.util.*;
import javax.swing.*;
import java.awt.event.*;

public class SprakShifteGUI extends JFrame{

private JPanel jPanel1 = new JPanel();
private JPanel inner_jPanel = new JPanel();
private JComboBox jComboBox1 = new JComboBox();
private JButton jButton1 = new JButton();
String jButtonSMStekst="";
String jButtonKodetekst="";
private JButton tilbakeButton1 = new JButton();
private JLabel headerLabel = new JLabel();
private JLabel innerHeaderLabel = new JLabel();
private JLabel aLabel = new JLabel();
private JLabel loggInnLabel = new JLabel();
private JLabel stegLabel = new JLabel();
private JLabel brukTextArea = new JLabel();
String brukSMSAlternativ="";
String brukKodeAlternativ="";
private JRadioButton smsRadio = new JRadioButton();
private JRadioButton engangskodeRadio = new JRadioButton();
private JLabel frnLabel = new JLabel();
private JTextField frnTF = new JTextField();
private JLabel passordLabel = new JLabel();
private JTextField passTF = new JTextField();
private JLabel andreLabel = new JLabel();

```

```

private JLabel bLabel = new JLabel();
private JLabel bLangLabel = new JLabel();
private JLabel cLabel = new JLabel();
private JLabel cLangLabel = new JLabel();
private JLabel lesLabel = new JLabel();
private JLabel hjelpLabel = new JLabel();

public SprakShifteGUI(){
setTitle("altinn - enklere rapportering til det offentlige");
setSize(600,600);
try{
jblnit();
}
catch(Exception e){
e.printStackTrace();
}
}

/*
 * INITAILISERER ELEMENTENE I GUI OG SETTER DEFAULT VERDI I FELTENE PÅ
BOKMÅL
 */
private void jblnit() throws Exception{
jPanel1.setLayout(null);
jPanel1.setSize(new Dimension(548, 586));
inner_jPanel.setLayout(null);
inner_jPanel.setBounds(new Rectangle(25, 55, 495, 500));
inner_jPanel.setBackground(Color.WHITE);
jComboBox1.setBounds(new Rectangle(335, 15, 185, 20));
String språk = "Norsk(Bokmål) - NB";
String språk2 = "Norsk(Nynorsk) - NN";
String språk3 = "Sami - SE";
String språk4 = "English - EN";
tilbakeButton1.setText("<< Tilbake til Altinn startside");
tilbakeButton1.setBackground(Color.GREEN);
tilbakeButton1.setBounds(new Rectangle(10, 450, 200, 23));
headerLabel.setText("<HTML><b>altinn - enklere rapportering til det offentlige</HTML></b>");
headerLabel.setBounds(new Rectangle(25, 10, 290, 30));
innerHeaderLabel.setText("<HTML><b>Velkommen til Altinn!</b></HTML>");
innerHeaderLabel.setBounds(new Rectangle(18, 10, 290, 30));
jComboBox1.addItem(språk);
jComboBox1.addItem(språk2);
jComboBox1.addItem(språk3);
jComboBox1.addItem(språk4);
jComboBox1.addActionListener(new ActionListener(){
public void actionPerformed(ActionEvent e){
jComboBox1_actionPerformed(e);
}
});
aLabel.setText("<HTML><B>A</B></HTML>");
aLabel.setBounds(new Rectangle(18, 50, 20, 30));
loggInnLabel.setText("Logg inn med passord:");
loggInnLabel.setBounds(new Rectangle(50, 50, 120, 30));
stegLabel.setText("Steg 1 av 2");
stegLabel.setBounds(new Rectangle(50, 80, 120, 30));
brukTextArea.setBounds(new Rectangle(50, 100, 220, 80));

```

```

brukSMSAlternativ="<HTML>Bruk ditt selvvalgte passord og klikk på Fortsett-knappen for å
bekrefte din identitet med engangskode.</HTML>";
brukKodeAlternativ="<HTML>Bruk ditt selvvalgte passord og klikk på Logg inn-
knappen.</HTML>";
brukTextArea.setText(brukSMSAlternativ);
smsRadio.setText("Send meg SMS med engangskode.");
smsRadio.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e){
        smsRadio_actionPerformed(e);
    }
});
smsRadio.setBounds(new Rectangle(50, 200, 220, 30));
smsRadio.setSelected(true);
engangskodeRadio.setText("<HTML>Logg inn uten engangskode.(gir begrenset
funksjonalitet)</HTML>");
engangskodeRadio.setBounds(new Rectangle(50, 230, 220, 30));
engangskodeRadio.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e){
        engangskodeRadio_actionPerformed(e);
    }
});
frnLabel.setText("Fødselsnummer");
frnLabel.setBounds(new Rectangle(50, 290, 150, 30));
frnTF.setText("");
frnTF.setBounds(new Rectangle(150, 290, 100, 25));
passordLabel.setText("Passord");
passordLabel.setBounds(new Rectangle(50, 320, 150, 30));
passTF.setText("");
passTF.setBounds(new Rectangle(150, 320, 100, 25));
jButtonSMStekst="Fortsett >>";
jButtonKodestekst="Logg inn >>";
jButton1.setText(jButtonSMStekst);
jButton1.setBounds(new Rectangle(150, 360, 140, 25));
jButton1.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e){
        jButton1_actionPerformed(e);
    }
});
andreLabel.setText("<HTML><B>Andre muligheter:</B></HTML>");
andreLabel.setBounds(new Rectangle(290, 70, 120, 30));
bLabel.setText("<HTML><B>B</B></HTML>");
bLabel.setBounds(new Rectangle(290, 110, 120, 30));
bLangLabel.setText("<HTML>Logg inn med engangskode. Dersom du har glemt passordet
eller ikke har registrert mobiltelefonnummer, kan du bruke engangskode (PINKode) tilsendt i
skattekort, selvangivelse eller Altinn-brev.</HTML>");
bLangLabel.setBounds(new Rectangle(310, 100, 170, 190));
cLabel.setText("<HTML><B>C</B></HTML>");
cLabel.setBounds(new Rectangle(290, 280, 120, 30));
cLangLabel.setText("<HTML>Logg inn med Buypass Smartkort. Dersom du innehar et
Buypass Smartkort kan du bruke dette for å logge inn i Altinn.</HTML>");
cLangLabel.setBounds(new Rectangle(310, 270, 170, 110));
lesLabel.setText("<HTML>Les om personvern i Altinn.</HTML>");
lesLabel.setBounds(new Rectangle(310, 360, 120, 60));
hjelpLabel.setText("<HTML>Hjelp til innlogging.</HTML>");
hjelpLabel.setBounds(new Rectangle(310, 410, 120, 30));
jPanel1.add(headerLabel, null);

```



```

jPanel1.add(jComboBox1, null);
inner_jPanel.add(jButton1, null);
inner_jPanel.add(tilbakeButton1, null);
inner_jPanel.add(innerHeaderLabel,null);
jPanel1.add(inner_jPanel, null);
inner_jPanel.add(aLabel);
inner_jPanel.add(loggInnLabel);
inner_jPanel.add(stegLabel);
inner_jPanel.add(brukTextArea);
inner_jPanel.add(smsRadio);
inner_jPanel.add(engangskodeRadio);
inner_jPanel.add(frnlLabel);
inner_jPanel.add(frnlTF);
inner_jPanel.add(passordLabel);
inner_jPanel.add(passTF);
inner_jPanel.add(andreLabel);
inner_jPanel.add(bLabel);
inner_jPanel.add(bLangLabel);
inner_jPanel.add(cLabel);
inner_jPanel.add(cLangLabel);
inner_jPanel.add(lesLabel);
inner_jPanel.add(hjelpLabel);
Container contentPane = getContentPane();
contentPane.add(jPanel1,null);
}

//GJØR INGENTING SIDEN DEN IKKE HAR NOE MED SPRÅKBYTTET Å GJØRE
private void jButton1_actionPerformed(ActionEvent e){
}

//UTFØRER KALL PÅ ØVERSTE RADIOKNAPP
private void smsRadio_actionPerformed(ActionEvent e){
    smsRadio.setSelected(true);
    engangskodeRadio.setSelected(false);
    brukTextArea.setText(brukSMSAlternativ);
    jButton1.setText(jButton1SMStekst);
}

//UTFØRER KALL PÅ NEDERSTE RADIOKNAPP
private void engangskodeRadio_actionPerformed(ActionEvent e){
    smsRadio.setSelected(false);
    engangskodeRadio.setSelected(true);
    brukTextArea.setText(brukKodeAlternativ);
    jButton1.setText(jButton1Kodetekst);
}

/*
 * KALLER METODEN lesSpraktabell() MED VALG FRA COMBOBOX
 * FÅR VECTOR MED SPRÅKFELTENE I RETUR
 * Fyller feltene med riktig språk
 */
private void jComboBox1_actionPerformed(ActionEvent e){
    int valgint = jComboBox1.getSelectedIndex();
    Vector spraktab=new Vector();
    spraktab=lesSpraktabell(valgint);
    jButton1SMStekst = (String)spraktab.get(11);
}

```

```

jButtonKodetekst =(String)spraktab.get(19);
headerLabel.setText((String)spraktab.get(0));
setTitle((String)spraktab.get(0));
innerHeaderLabel.setText((String)spraktab.get(1));
loggInnLabel.setText((String)spraktab.get(2));
stegLabel.setText((String)spraktab.get(3));
brukSMSAlternativ=(String)spraktab.get(4);
brukKodeAlternativ=(String)spraktab.get(18);
brukTextArea.setText(brukSMSAlternativ);
smsRadio.setText((String)spraktab.get(5));
engangskodeRadio.setText((String)spraktab.get(6));
frnLabel.setText((String)spraktab.get(7));
tilbakeButton1.setText((String)spraktab.get(17));
passordLabel.setText((String)spraktab.get(9));
jButton1.setText(jButtonSMStekst);
andreLabel.setText((String)spraktab.get(12));
bLangLabel.setText((String)spraktab.get(13));
cLangLabel.setText((String)spraktab.get(14));
lesLabel.setText((String)spraktab.get(15));
hjelpLabel.setText((String)spraktab.get(16));
smsRadio.setSelected(true);
engangskodeRadio.setSelected(false);
}

//LESER SPRÅKTABELL FRA FILEN SPRÅKTAB.TXT
public Vector lesSpraktabell(int valg){
    int sprakvalg = valg+1;
    Vector sprak = new Vector();
    String filnavn = "spraktab.txt";
    try{
        FileReader leseforbTilFil = new FileReader(filnavn);
        BufferedReader leser = new BufferedReader(leseforbTilFil);
        String linje = leser.readLine();
        while(linje!=null){
            StringTokenizer tekst = new StringTokenizer(linje,"t");
            int counter=0;
            String inn= tekst.nextToken();
            while(counter<=sprakvalg){
                if((counter)==(sprakvalg)){
                    sprak.add(inn);
                }
                if(tekst.hasMoreTokens()){
                    inn= tekst.nextToken();
                }
                counter++;
            }
            linje = leser.readLine();
        }
    }catch(IOException io){
        System.out.println("Error with reading file" + filnavn + "\n"+io);
    }
    return sprak;
}
}

package language;

```

```

import java.sql.Timestamp;
import java.util.*;
import javax.swing.JFrame;

class RunSprak
{
    public static void main(String[] args)
    {

        JFrame frame = new SprakShifteGUI();
        frame.setSize(600,600);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.show();
        SprakShifteGUI ss = new SprakShifteGUI();

    }
}

```

Programkode 6. Klassen SprakShifteGUI. Logg inn siden hos Altinn med språkbytte programmert i Java. Klassen RunSprak nederst starter programmet.

Felttabell

1	altinn – enklere rapportering til det offentlige	altinn – enklere rapportering til det offentlige	altinn – enklere rapportering til det offentlige	altinn – enklere rapportering til det offentlige
2	<HTML>Velkommen til Altinn!</HTML>	<HTML>Bures boahhtin Altinnai!</HTML>	<HTML>Velkommen til Altinn!</HTML>	<HTML>Welcome to Altinn!</HTML>
3	<HTML>Logg inn med passord:</HTML>	<HTML>Čálit sisa čovddasániin:</HTML>	<HTML>Logg inn med passord:</HTML>	<HTML>Log in with password:</HTML>
4	<HTML>Steg 1 av 2</HTML>	<HTML>Step 1 of 2</HTML>	<HTML>Steg 1 av 2</HTML>	<HTML>1. oassi 2 oasis</HTML>
5	<HTML>Bruk ditt selvvalgte passord og klikk på Fortsett-knappen for å bekrefte din identitet med engangskode.</HTML>	<HTML>Čovddasáni ja deaddel Joatkke boalu duođastan dihte iežat identitehta háválás čovddasániin.</HTML>	<HTML>Bruk ditt selvvalgte passord og klikk på Fortsett-knappen for å bekrefte din identitet med engangskode.</HTML>	<HTML>Ane iešválljejuvvon doaimmat.</HTML>
6	<HTML>Send meg SMS med engangskode</HTML>	<HTML>Sádde munnje háválás čovddasáni SMS bokte</HTML>	<HTML>Send meg SMS med engangskode</HTML>	<HTML>Send me SMS with one use PIN code</HTML>
7	<HTML>Logg inn uten engangskode. (gir begrenset funksjonalitet)</HTML>	<HTML>Logg inn uten engangskode. (gir begrenset funksjonalitet)</HTML>	<HTML>Logg inn uten engangskode. (gir begrenset funksjonalitet)</HTML>	<HTML>Logg inn uten engangskode. (gir begrenset funksjonalitet)</HTML>
8	<HTML>Fødselsnummer</HTML>	<HTML>Riegádannummar</HTML>	<HTML>Fødselsnummer</HTML>	<HTML>Fødselsnummer</HTML>
9				
10	<HTML>Passord</HTML>	<HTML>Čovddasátni</HTML>	<HTML>Passord</HTML>	<HTML>Password</HTML>
11				
12	Fortsett >>	Fortsett >>	Joatkke >>	Continue >>
13	<HTML>Andre muligheter:</HTML>	<HTML>Eará vejolašvuodát:</HTML>	<HTML>Andre muligheter:</HTML>	<HTML>Other Options:</HTML>

14 <HTML>Logg inn med engangskode. Dersom du har glemt passordet eller ikke har registrert mobiltelefonnummer, kan du bruke engangskode (PINkode) tilsendt i skattekort, selvangivelse eller Altinnbrev.</HTML> <HTML>Logg inn med engangskode. Dersom du har glemt passordet eller ikke har registrert mobiltelefonnummer, kan du bruke engangskode (PINkode) tilsendt i skattekort, selvangivelse eller Altinnbrev.</HTML> <HTML>Čálit sisa háválás čovddasániin. Jus leat vajálduhtán čovddasáni dahje it leat registrerejuvvon mobiltelefonnummar, de sáhtát atnit háválás čovddasáni mii čuožžu vearrokoarttas, iešdieđáhusas dahje Altinn reivves.</HTML> <HTML>Log in with one use code. If you have lost your password or haven't registered a cellular phonenumber, You may use a one time code (PINcode) recieved in Tax Card, Tax report or letter from Altinn.</HTML>

15 <HTML>Logg inn med Buypass Smartkort. Dersom du innehar et Buypass Smartkort kan du bruke dette for å logge deg inn i Altinn.</HTML> <HTML>Logg inn med Buypass Smartkort. Dersom du innehar et Buypass Smartkort kan du bruke dette for å logge deg inn i Altinn.</HTML> <HTML>Čálit sisa Buypass Smartkortain. Jus dus lea Buypass Smartkort de sáhtát dan atnit go čálihat sisa Altinnai.</HTML> <HTML>Logg in with Buypass Smartcard. If you possess a Buypass Smartkort, you may use this to log in to Altinn.</HTML>

16 <HTML>Les om personvern i Altinn.</HTML> <HTML>Les om personvern i Altinn.</HTML> <HTML>Loga personsuodjalusa birra Altinnas.</HTML> <HTML>Read up on personal rights in Altinn.</HTML>

17 <HTML>Hjelp til innlogging.</HTML> <HTML>Hjelp til innlogging.</HTML> <HTML>Sisačálihanveahkki.</HTML> <HTML>Help with log in</HTML>

18 << Tilbake til Altinn startside << Tilbake til Altinn startside << Ruovttoluotta Altinna álgosiidui << Back to Altin mainpage

19 <HTML>Bruk ditt selvvalgte passord og klikk på Logg inn-knappen.</HTML> <HTML>Bruk ditt selvvalgte passord og klikk på Logg inn-knappen.</HTML> <HTML>Ane iešválljejuvvon čovddasáni ja deaddel Čálit sisa boalu.</HTML> <HTML>Use your chosen password and click on the Log in-button</HTML>

20 <HTML>Logg inn</HTML> <HTML>Logg inn</HTML> <HTML>Čálit sisa</HTML> <HTML>Logg in</HTML>

spraktab.txt. Språktabell med alle felt fra Logg inn siden hos Altinn på bokmål, nynorsk, samisk og engelsk i den rekkefølgen. Det er benyttet TAB for å skille mellom språkene. For å gjøre filen lesbar er det mulig å åpne og konvertere den i for eksempel Excel

Vedlegg B: Referanser

Acharya (2002, 28. januar). *Multilingual Systems [online]* – URL: http://www.indictrans.org/Articles/English/article_src/Indic/rkk1.html og <http://acharya.iitm.ac.in/>

Babel Fish [online] – URL: <http://babelfish.altavista.com/translate.dyn>

Caplex nettleksikon. [online] - URL: <http://www.caplex.no/>

Målloven. Lov 11. April 1980 om målbruk i offentlig teneste med endringar av 11. mars 1988.

Altinn. *Om Altinn [online]*. – URL: <https://www.altinn.no/>

Norsk Språkråd (juni 2001). *Handlingsplan for norsk språk og IKT [online]*. – URL: <http://www.sprakradet.no/>

Sametingets språkavdeling [online] – URL: <http://www.samisk-sr.no>

Statistisk sentralbyrå [online] – URL: <http://www.ssb.no>

Trosterud, Trond (2002, 5. september). *Samisk på data [online]*. – URL: <http://www.hum.uit.no/a/trond/isit.html>

Unicode [online] – URL: <http://www.Unicode.org/>

ISO, *Codes for the Representation of Names of Languages [online]* – URL: <http://www.loc.gov/standards/iso639-2/englangn.html>

Wærås, Torgunn (2003, 24. mai). *Flerspråklig i en fei [online]*. – URL: <http://uit.no/nyheter/tromsoflaket/732>

Canada.ca [online] – URL: <http://canada.gc.ca/>

IGLO prosjektet [online] – URL: <http://tea.fernuni-hagen.de/Iglo/Install/kurs/Ressourcen/start.htm>

Nyno [online] – URL: <http://www.nyno.no>

Norge.no [online] – URL: <http://www.norge.no/>

Husand, Hallstein (2005, 14. mars) *Altinn – en nøkkel til et Norge i endring [online]* – URL: <http://www.samit.no/>

Wood, Allan (2005, 14. april) *Unicode and Multilingual Support in HTML, Fonts, Web Browsers and Other Applications* [online] – URL: <http://www.alanwood.net>

Altinn, *Elektronisk innrapporteringskanal for næringslivet, Implementasjonsguide*