

Visualisering av gassnettverket

Utarbeidet av:

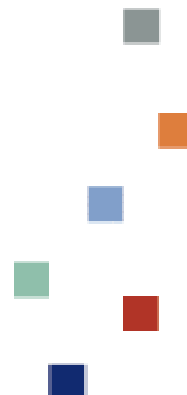
Anders Løken, Terje André Olsen

Fag:

Masteroppgave i informatikk

Avdeling:

Avdeling for informasjonsteknologi, 2009



Høgskolen i Østfold



www.hiof.no

Sammendrag

Prosjektet baserer seg på bruk av standard utviklingsverktøy for å visualisere gassnettverket i Europa.

Oppdragsgiver er Navita. Målet har vært å teste ut forskjellige teknologier og visualiseringsteknikker, med tanke på vedlikeholdbarhet og brukbarhet for forskjellige brukergrupper.

Hovedoppgavene våre har vært å teste og dokumentere mulighetene til Chart FX Maps og Google Maps for visualisering av kraftnettverket på ulike kartgrunnlag. For og kunne teste og dokumentere mulighetene har vi implementert en rekke forslag til løsninger som har blitt presentert og diskutert med oppdragsgiver regelmessig.

Dette dokumentet er bygget opp slik at vi starter med oppgavebeskrivelsen, hvor vi diskuterer oppgavebeskrivelsen og hvilke endringer som ble på gjort i samråd med oppdragsgiver. Videre beskriver vi hvilke utfordringer vi har arbeidet med i prosjektet. Neste kapittel inneholder bakgrunnsstoff og teori om de ulike temaer som er omtalt i dokumentet. I kapitlet med navn “Brukerundersøkelse” beskriver vi hvordan møtet med potensielle brukere av løsningen oppdragsgiver skal levere. Videre avslutter vi med “Teknisk beskrivelse” hvor det tekniske bidraget blir beskrevet, de to siste kapitlene omhandler konklusjon og fremtidig arbeid.

Nøkkelord:

Visualisering, kraftmegling, kart, XML

Forord

Denne masteroppgaven har blitt utarbeidet i felleskap og vi ønsker begge å takke Navita med Svein Olsen i spissen for muligheten til å utføre oppgaven, Navita har brukt tid og ressurser på å besvare våre spørsmål og gi oss nyttige råd og tips underveis. Vi er også takknemmelige for den friheten vi hadde med tekniske valg. Navitas fagfelt er meget interessant og vi har vært gjennom en svært lærerik prosess.

Vi ønsker å takke vår veileder på Høgskolen i Østfold, Børre Stenseth. Børre har vært en uvurderlig støttespiller gjennom hele prosjektiden.

Anders Løken og Terje Olsen

Rygge 20. januar 2009

Innhold

Sammendrag	1
Forord	2
1 Oppgavebeskrivelse	6
1.1 Oppgavebeskrivelse fra Navita	6
1.2 Revidert oppgave	7
1.2.1 Inndeling av oppgaven	7
2 Utfordringer	11
2.1 Målgruppe	11
2.1.1 Programvareutviklere	12
2.1.2 Sluttbrukere	12
2.2 Visualisering	12
2.3 Datagrunnlag	13
2.3.1 Nord Pool Spot prisinformasjon	14
2.3.2 Lagring av testdata	15
2.4 Applikasjonstype	17
2.4.1 Skrivebordsløsning	17
2.4.2 Webløsning	18
3 Bakgrunn	19
3.1 Børsbasert Krafthandel	19
3.2 Litt historikk om Nord Pool	20
3.3 Litt historikk om Navita	20
3.4 Visualiseringsteori	21
3.4.1 Transportplan for fly	21
3.4.2 Visualisering av flyt	23
3.4.3 Metrokart, London Underground	24
3.4.4 Geografisk korrekte løsninger	28
3.4.5 Bruk av farger	31
3.4.6 Detaljeringsgrad	33
3.4.7 Dynamisk oppdatering	35

4	Teknisk bakgrunn	36
4.1	Verktøy og standarder	36
4.1.1	Visual Studio.net og Chart FX Maps	36
4.1.2	Extensible Markup Language	39
4.1.3	XML-skjema	39
4.1.4	Scalable Vector Graphics	40
4.1.5	Google Maps	40
4.1.6	KML	41
4.1.7	Aspx.net	41
4.1.8	Geonames	42
4.1.9	JSON	42
4.2	Utfordringer og forslag til løsninger	42
4.2.1	Inndeling av prissoner	42
4.2.2	Bruk av løsninger på tvers av teknologiene	46
5	Brukerundersøkelse	50
5.0.3	Halden Center of Expertise Energy Research	50
5.0.4	Metode	51
6	Tekniske løsninger	57
6.1	Skrivebordsløsning	57
6.2	Webløsning	61
6.2.1	Inndeling av prissoner	62
6.2.2	Visualisere prissoner	63
6.2.3	Generelle prissoner	64
6.2.4	Visualisere rørledninger	67
7	Konklusjon	70
7.1	Skrivebordsløsning	70
7.1.1	Visualisere prissoner på kartgrunnlag	71
7.1.2	Visualisere tidsperspektiv på forskjellige prisområder	71
7.1.3	Mulighet for zoom	71
7.1.4	Visualisere rørledninger, noder med logistikk informasjon	71
7.1.5	Flyt mellom prissoner	72
7.1.6	Visualisere smart-mapping, metrokart	72
7.1.7	Sammendrag skrivebordsløsning	73
7.2	Webløsning	74
7.2.1	Visualisere prissoner på kartgrunnlag	74
7.2.2	Visualisere tidsperspektiv på forskjellige prisområder	75
7.2.3	Mulighet for zoom	75
7.2.4	Visualisere rørledninger, noder med logistikk info	75
7.2.5	Flyt mellom prissoner	76
7.2.6	Visualisere smart-mapping, metrokart	76
7.2.7	Sammendrag webløsning	76
7.3	Oppsummering	77

8 Fremtidig Arbeid	79
8.0.1 Google Traffic Information	80
Bibliografi	82
Liste over figurer	87
Liste over tabeller	88
A Delivery	89
A.1 Desktop application	89
A.1.1 Nordpoolspot	89
A.2 Web application	90
A.2.1 Circleoverlay	90
A.2.2 Gaspipes	90
A.2.3 Nordpoolspot	91
A.2.4 Overlaymaking	91
B Demonstrasjon av webløsning	92
B.1 Circleoverlay	92
B.2 Gaspipes	92
B.3 Nordpoolspot	92
B.4 Overlaymaking	92

Kapittel 1

Oppgavebeskrivelse

Denne masteroppgaven er utarbeidet i samarbeid med Navita. Navita leverer løsninger til bedrifter som driver med kjøp og salg av blant annet elektrisk kraft, gass, kull, CO₂kvoter og lignende. Den originale oppgavebeskrivelsen fra Navita er gjengitt nedenfor. Oppgavebeskrivelsen har i samarbeid med Navita blitt revidert flere ganger.

1.1 Oppgavebeskrivelse fra Navita

Visualisering av nettstruktur

“Elektrisitetsmarkedene og gassmarkedene i Europa er gjennom dereguleringsprosessen blitt åpent for flere aktører enn de tidligere monopolistene. Et av verktøyene for å gjennomføre denne prosessen har vært et klart skille mellom de som opererer rørledninger og ledningsnett, og de som er kommersielle aktører som aktivt handler og forsyner gass og elektrisitet. For slike kommersielle aktører blir evnen til å ha full oversikt over hvor de har kjøpt/produserer gass eller kraft og hvor de har forpliktelser essensielt, men ikke minst hvordan de evner og planlegger å transportere fra ett sted til et annet. Systemer for å håndtere disse logistikkutfordringene er ett av hovedsatsningsområdene til Navita.

Elektrisitets- og gassnettverket består av elementer (f. eks. generator, transformator) og kobling(er) mellom disse. Videre er det geografiske områder som inneholder flere tilkoblingspunkter som har felles status (f. eks. prisområde, nettverksområde). En generell utfordring har alltid vært å visualisere en aktørs posisjoner inn i det fysiske nettverket, det finnes så vidt oss bekjent ikke noen virkelig gode løsninger på dette problemet lett tilgjengelig i markedet.

Navitas forslag til oppgave er å visualisere dette ved bruk av en standard 3 part komponent, Chart FX Maps. Visualisering må kunne vises geografisk korrekt i tillegg til å kunne ha en visuell lesbar korrigerings (f. eks. slik som er gjort for Londons undergrunnskart). Videre vil vi ha undersøkt muligheten for å kunne bruke internettbaserte geografiske kartprogrammer som f. eks. Microsoft Virtual Earth eller Google Earth, for å kunne vise nettverkstrukturen i et mer

detaljert og kraftigere verktøy. Dette siste ser vi for oss er en tilleggsfunksjonalitet hvis bruksverdi nærmest bare kan avdekkes gjennom eksperimentering slik vi legger opp til her.

Vi er åpne for justeringer av tilnærming, og har en del ytterligere informasjon/ideer tilgjengelig dersom dette er ønskelig.”

-Svein Olsen CTO Navita.

1.2 Revidert oppgave

Navita ønsket med oppgaven å finne egnet teknologi, med tanke på framstilling og dataregistrering. De fire punktene vi har gjengitt nedenfor har til stor grad vært vår arbeidsform.

- Identifisere behov, fastsette krav
- Utvikle flere typer design
- Lage interaktive prototyper
- Evaluere design

Vi startet med å definere hver underoppgave sammen med Navita. Designet på løsningen laget vi i henhold til hvilke ønsker Navita ga uttrykk for. Til neste møte med Navita laget vi et løsningsforslag, der vi i samarbeid med Navita evaluerte løsningen. Neste steg i prosessen er å forbedre løsningsforslaget, eller endre design og/eller teknologi. På denne måten har rammene for arbeidet vært ganske dynamisk. En slik iterativ arbeidsform er utfordrende, men interessant.

1.2.1 Inndeling av oppgaven

Gjennom ulike revisjoner har oppgaven fått en todeling:

- Skrivebordsløsning
- Webløsning

Det finnes fordeler og ulemper ved begge plattformer. Disse egenskapene er generelle og ikke spesifikke for denne oppgaven. En av fordelene ved en skrivebordsapplikasjon er tilgjengelighet. Det vil si at en skrivebordsapplikasjon er uavhengig at en webserver er tilgjengelig. En ulempe med skrivebordsapplikasjoner er installasjonsprosessen til hver enkelt klient. Dette er en jobb som øker i omfang samtidig som brukermassen øker. Videre må man tenke på at eventuelle oppdateringer og oppgraderinger av programmet krever at bruker eller it-konsulenter må involveres. Ulempene som vi har nevnt ved en skrivebordsapplikasjon blir eliminert ved bruk av webbaserte applikasjoner. Ingen, eller svært lite arbeid kreves ved installasjon på klientene kreves. Ulempene er selvfølgelig avhengigheten av at webserveren er tilgjengelig.

I hovedsak var det skrivebordsløsningen som skulle utvikles, men grunnet tekniske mangler på utviklingsverktøyet fikk utviklingsprosessen en naturlig overgang til et nytt utviklingsmiljø. I samtale med Navita ble vi enige om å utvikle en løsning ved bruk av internettbaserte geografiske kartprogrammer. Valg som blir gjort underveis er ikke nødvendigvis de valg vi ville tatt selv, men vi vil drøfte positive og negative sider ved de ulike valg som blir gjort.

Målet med begge løsningene er å implementere støtte for disse nøkkelfunksjonene:

1. Visualisere prissoner på kartgrunnlag.
2. Visualisere tidsperspektiv på forskjellige prisområder.
3. Ha mulighet for å zoome til prisområder.
4. Visualisere rørledninger, noder med logistikk informasjon.
5. Vise flyt av kraft mellom prissoner.
6. Visualisere smart-mapping, “metrokart”.

Skrivebordsløsning

Denne oppgaven går ut på å visualisere gassnettverket på et europakart. Her vil det bli lagt vekt på korrekt geografisk fremstilling. Informasjon som skal være med i denne oppgaven er inndeling av prissoner, flyt av kraft mellom soner, ledningsnett og gassinstallasjoner (knutepunkter og noder). Hvert land har som regel en prissoner. Norge skiller seg ut ved å ha tre dynamiske prissoner. Problemet med dynamiske prissoner setter spesielle krav til løsningen, spesielt datastrukturen. Prissonene må kunne settes sammen dynamisk av mindre geografiske områder, typisk kommunenivå.

Figur 1.1 viser soneinndeling, prisinformasjon og flyt av kraft. “NordPool Spot” er en fremstilling av denne informasjonen dynamisk, basert på tid og/eller årstidene. Dette er ikke et fullt utviklet system for sluttbrukere. Men fungerer som en illustrasjon. Vi skal undersøke muligheten for å benytte moderne hyllevare utviklingsverktøy og standarder for å lage et system til sluttbrukerne.

Løsningen ble ønsket basert på .net fra Microsoft. Fremstillingen skulle gjøres i Chart FX Maps.[11] Det finnes en gjennomgang av verktøy og standarder brukt i oppgaven i seksjon 4.1.



Figur 1.1: Illustrasjon av Nord Pool Spot

Det har blitt endringer i arbeidsoppgaver underveis. Krav og funksjoner har blitt justert fortløpende. Hovedgrunnen til dette er at dette er utforskningsprosjekt der vi har blitt forespurt å finne ut om dette lar seg gjøre innenfor de rammene vi har blitt gitt. En av rammene er teknologivalg, der Navita ønsker å teste ut de tekniske løsningene som de har valgt til dette prosjektet. Valg av teknologi og løsninger er en viktig del av denne masteroppgaven. Gjennomgang av tekniske formater og teknologier har blitt forklart generelt i seksjon 4.1. Hvordan vi har benyttet teknologien og våre resultater beskrives i seksjon 4.2.

For å få en så korrekt geografisk fremstilling som mulig så har vi valgt å benytte GPS-koordinater. Filosofien bak bruken av GPS-koordinater gjør at vi får et koordinatsystem som er universelt. Det er vanlig i kartsammenheng å bruke lengde- og breddegrader for å henvise til et punkt på kartet. Bruken av disse koordinatene vil gjøre ting enklere i etterkant når nye elementer skal plasseres geografisk korrekt. Under prosjektets gang har det blitt utviklet en metode for å oversette Chart FX compatible SVG-kart fra dets eksisterende koordinater til GPS-koordinater. Dette er fordi at vi ønsker å “generalisere” kartene, slik at løsningene kan benyttes med andre løsninger som benytter GPS koordinater. Kartet vi har benyttet oss av i skrivebordsløsningen er laget av Navita, og baserer seg på samme oppsett som kartene som følger med Maps tileggspakken til Chart FX. Disse kartene benytter seg av relative pixel-koordinater. Det er problematisk å benytte overlays som er laget med et annet koordinatsystem. Vi gjorde et forsøk på å konvertere det relative koordinatsystemet til et GPS basert koordinatsystem. Teorien bak GPS konvertering og metoden for konvertering står beskrevet i seksjon 4.2.2.

Oppsummering

Kort fortalt skal skrivebordsløsningen ha denne funksjonaliteten:

- prissoner på kartgrunnlag.
- Tidsperspektiv på forskjellige prisområder.
- Ha mulighet for å zoome til prisområder.
- Rørledninger, noder med logistikk informasjon.
- Vise flyt mellom prissoner.
- Smart-mapping, “metrokart”.

Webløsning

Denne oppgaven har de samme funksjonelle krav som skrivebordsløsningen. Navita har i oppgaveteksten foreslått Microsoft Live Earth og Google Earth. Vi ville i stedet se på muligheten for å benytte webleseren som omgivelse som fortsatt var Microsoft Live Earth og Google Earth. Selv om disse inneholder mye funksjonalitet, ser vi det som unødvendig komplisert å skreddersy disse løsningene tilstrekkelig til at de kan brukes i kommersiell sammenheng.

Dersom vi benytter Google Maps som det finnes en tilgjengelig API for, mener vi at er det større muligheter for å tilpasse løsningene etter Navitas ønske. De funksjonelle kravene til webløsningen er de samme som for skrivebordsløsningen. Dette er en fordel for oss, da vi kan benytte oss av kunnskaper som vi tilegnet oss under utviklingen av skrivebordsløsningen. Navita ytret et ønske om at vi skulle bruke Microsoft sine produkter, men dette er ikke et krav. Valget av hvilken løsning vi ville bruke ble tatt på bakgrunn av tilgjengelig dokumentasjon og materiale. Det finnes mye informasjon tilgjengelig på hvordan Google Maps API kan brukes som verktøy til å visualisere ulike lokasjonsbaserte problemstillinger. En av kildene som ble brukt for å se på muligheter og for å finne inspirasjon for å jobbe med Google Maps er bloggen Google Mapsmania[4] og boken “Beginning Google Maps Applications with Rails and Ajax”[21] som er en samleside for ulike ikke-kommersielle Google Maps prosjekter. Mer om valg av teknologi finnes i seksjon 4.1.

Skrivebords- og webløsningen har de samme kravene til funksjonalitet og visualisering. Datastrukturen og datagrunnlaget er ment skal være det samme ved bruk av Navitas XML-skjemabeskrivelse (XSD). Forskjellen ligger i teknologivalg og applikasjonstype. Derfor gjelder underoppgavene til skrivebordsløsningen også for webløsningen.

Kapittel 2

Utfordringer

Som nevnt i oppgavebeskrivelsen blir denne oppgaven utført i et samarbeid med Høgskolen i Østfold og Navita, som har utarbeidet oppgaveteksten og er oppdragsgivere for dette prosjektet. Vi ble enige med Navita om hyppige møter for å diskutere våre forslag til løsninger, og videre fremdrift. I løpet av en periode på 9 måneder har vi hatt 13 møter med Navita. Vår veileder på skolen har hjulpet oss med utformingen av masterrapporten og med teknologiske og strategiske valg.

2.1 Målgruppe

Opgaven vår er i hovedsak å komme fram til forslag på løsninger på oppgavene som blir beskrevet i oppgavebeskrivelsen. Resultatet av vårt arbeid blir overlevert til Navita, derfor kan vi si at hovedmålgruppen for programvaren vi utvikler er Navitas programvareutviklere. Navita planlegger å videreutvikle våre forslag til løsninger basert på vårt arbeid dersom de finner dette hensiktsmessig. Dette prosjektet letter arbeidet til Navitas programvareutviklere ved å prøve ut ulike teknologier. Slik utprøving er tidkrevende, og alle muligheter som vi har testet er tid spart for Navitas utviklere. Siden framstilling/GUI står sentralt i vår oppgave var det nødvendig å få vurdert våre løsninger med tanke på fremtidige brukere.

Det er viktig å understreke hva vi definerer som vår målgruppe. Navitas programvareutviklere er vår primær-målgruppe som skal motta vår leveranse. I tillegg til dette refererer vi til sluttbrukerne av løsningen. Dette er fordi vi skal få frem funksjonalitet og krav som disse brukerne stiller. Våre valg med tanke på grensesnitt og utseende kan endres av Navitas utviklere, men vi ønsker å gjøre denne jobben så enkel som mulig for utviklerne, da vi har beskrevet hvilke krav og funksjonalitet som er viktig for de daglige brukerne av løsningen.

Det er denne målgruppen som skal bruke applikasjonen daglig, og vil derfor bli tilpasset deres behov. De ulike målgruppene ble fastsatt ved å intervjuer en faggruppe som har kjennskap til de mulige brukerne og deres arbeidsoppgaver.

Dersom våre forslag er i forhold til Navitas ønsker om utviklingstid, kostnader og forventet resultat kan avgjørelsen om å videreutvikle våre løsninger til et fullverdig system bli tatt. Det vil være flere brukergrupper av løsningen, og vi har sett på et utvalg av brukere. Vi vil også beskrive de forskjellige brukergruppene og deres krav til løsninger.

Det er nyttig for oss å tenke på sluttbrukere. Disse kan gi oss en mer reell arbeidssituasjon der brukernes krav og ønsker blir ivaretatt helt fra tidlige stadier i planleggingen. Når brukerne har vært med på hele prosessen kan vi ende opp med ett bedre sluttresultat. Å involvere brukere tidlig i designfasen er en av interaksjonsdesign viktigste byggesteiner. De teknologiene som velges har konsekvenser på hvordan løsninger kan utformes for sluttbrukeren.

2.1.1 Programvareutviklere

Som tidligere nevnt har vi forskjellige brukere vi må forholde oss til. Vi må tenke på sluttbrukernes ønsker til løsningene, men det er Navitas programvareutviklere som er vår primære målgruppe. Disse utviklerne får overlevert våre forslag til løsninger. Det er et krav at disse forslagene er laget i de formater og teknologier som vi har blitt enige om. Samtidig skal vi overlevere forklart kode og oversikt over løsninger. Vår oppgave er å undersøke mulighetene i teknologien, mens Navitas programvareutviklere skal videreutvikle løsningene til å passe sluttbrukernes ønsker. Det er derfor viktig for oss å ha et klart bilde også av sluttbrukernes behov.

2.1.2 Sluttbrukere

Vi har identifisert ulike brukergrupper for programmet:

- **Kraftmegler**, hvor de viktigste kravene er: Prisinformasjon, lettlest grafikk, god oversikt, og raske oppdateringer.
- **Forsker/Analytiker**, hvor de viktigste kravene er: Detaljert informasjon, historikk, trender, nøyaktighet.
- **Nettoperatør**, hvor de viktigste kravene er logistikkinformasjon: Kostnader, inntekter, gasstrykk, flyt. etc.

2.2 Visualisering

En av de viktigste utfordringene i denne oppgaven er visualisering. Navita ønsker å se på muligheter for forskjellige typer visualisering. Forslag som Navita har kommet med i tillegg til geografisk korrekthet er en tankegang med bakgrunn i “metrokart”. De fleste kart av verdens undergrunnsbaner har kopiert prinsippene fra Londons undergrunnsbane, eller deler av den. I korthet kan vi forklare dette med å endre på geografiske lokasjoner, for å oppnå bedre oversikt og lesbarhet.

Vi har sett på ulike teknikker for å visualisere transport problematikk, som ruteplan for en flyplass og undergrunnskart med hovedvekt på originalen; London Underground i seksjon 3.4.3.

Navita ønsket at vi skulle se litt på hvilke muligheter det ligger i teknologien for disse utfordringene. Vi startet med å visualisere med geografisk korrekthet. Vi har laget løsningsforslag for forskjellige applikasjonstyper, skrivebordsløsning og webløsning. Tiden det tok å lage disse forslagene var omfattende. Dette gjorde at vi i samarbeid med Navita konsentrerte oss om å få ferdig forslag for geografisk korrekthet.

Det er en utfordring å implementere forslagene til Navita, da ingen av partene er helt sikre på hvordan dette skal gjøres, og hvordan sluttresultatet kan eller bør bli.

2.3 Datagrunnlag

En av de sentrale delene i begge løsningene er å presentere data. Navita har systemer for å hente de data vi trenger. Denne informasjonen kan deles mellom systemene, da Navita bruker XML filer til sin løsning. Dersom vi benytter XML filer vil vi kunne oppnå en dynamikk i våre løsninger som gjør at all informasjon som oppdateres i Navitas system også vil oppdatere informasjonen i våre løsninger. På denne måten vil vår data til enhver tid være den samme som i Navitas system. For at denne løsningen skal fungere er vi nødt til at XML filene har lik oppbygging eller at de lett kan transformeres f. eks ved hjelp av XSLT. Dette kan vi forsikre oss om ved å benytte like XML-skjema (XSD) filer. Skjemafilen kan brukes til å validere en XML-fil og inneholder regler om hvordan XML filen skal bygges opp. Skjemafilen som trengs for å samarbeide med Navitas system er ikke tilgjengelig i sin endelige form.

Vi var derfor nødt til å finne en midlertidig løsning for å lagre prisinformasjon og geografiske data. Vi brukte XML filer for dette, og tankegangen var å tilpasse løsningene og XML filen til XSD filene når disse ble gjort tilgjengelig. De midlertidige XML filene våre viste seg å bli de endelige XML filene. Innhenting og lagring av testdata, samt lagring av geografiske data er beskrevet nedenfor.

Uavhengig av hvilken visualiseringsstrategi vi velger trenger vi testdata i våre løsninger. En av byggeklossene i en slik løsning vi jobber med å utvikle er korrekte data. Formålet med løsningen er å presentere data, dersom data er ukorrekt er ikke denne løsningen brukbar for noen av de ulike brukergruppene. I den ferdige løsningen vil Navita hente oppdaterte data fra eksisterende løsninger, men vi valgte å operere med så reelle tall som mulig for at vi skal få et bedre innblikk i hvordan kraftprisene fungerer. Vi har jobbet tett opp mot Nord Pool Spot (Figur 3.1) som tilbyr oppdaterte lister over priser for alle områder som er med i den nordiske kraftbørsen.

Under følger en illustrasjon om hvordan output fra Nord Pool Spot[9] er:

2.3.1 Nord Pool Spot prisinformasjon

Date Thu 01.05.08

Time

	SYS	DK1	DK2	NO1	NO2	NO3	FI	SE
00-01	13,07	29,35	10,02	10,02	10,02	39,91	10,02	10,02
01-02	8,74	13,42	3,93	8,86	3,93	33,95	3,93	3,93
02-03	4,55	6,68	3,79	6,68	3,79	33,92	3,79	3,79
03-04	3,79	4,53	3,11	4,53	3,11	21,40	3,11	3,11
04-05	3,76	5,24	2,59	5,24	2,59	2,59	2,59	2,59
05-06	3,19	5,44	1,71	5,44	3,80	3,80	1,71	1,71
06-07	3,49	7,29	1,72	7,29	3,82	3,82	1,72	1,72
07-08	8,12	9,42	3,08	9,42	3,86	33,94	3,08	3,08
08-09	17,06	30,02	30,02	15,04	15,04	15,04	15,04	15,04
09-10	17,89	42,38	42,38	16,69	16,69	16,69	16,69	16,69
10-11	18,62	42,84	42,84	17,25	17,25	17,25	17,25	17,25
11-12	19,12	43,08	43,08	17,56	17,56	39,91	17,56	17,56
12-13	18,65	46,41	46,41	17,39	17,39	39,90	17,39	17,39
13-14	17,93	37,68	37,68	16,96	16,96	33,99	16,96	16,96
14-15	17,83	31,07	31,07	16,73	16,73	33,97	16,73	16,73
15-16	17,45	29,56	29,56	16,52	16,52	39,91	16,52	16,52
16-17	17,52	29,78	29,78	16,58	16,58	33,98	16,58	16,58
17-18	17,91	38,92	38,92	16,76	16,76	33,96	16,76	16,76
18-19	18,32	46,58	46,58	17,02	17,02	33,96	17,02	17,02
19-20	18,82	45,70	45,70	17,36	18,02	33,98	18,02	18,02
20-21	20,29	52,77	52,77	17,56	22,34	33,95	22,34	22,34
21-22	22,31	54,94	54,94	17,88	28,32	39,93	28,32	28,32
22-23	22,01	56,03	56,03	17,71	28,93	39,97	28,93	28,93
23-24	20,40	50,48	50,48	17,41	28,72	39,94	28,72	28,72

Forklaring av figur:

Kolonne 1 viser hvilken time av døgnet den aktuelle prisen gjelder for.

Kolonne 2 inneholder "System Price" som er en avregning basert på hele Norden. Avregningen er krysningspunktet mellom tilbud og etterspørsel av kraft i Norden.[8] De resterende kolonner inneholder prisinformasjon til en bestemt prissone:

DK1, DK2 = Danmark

NO1, NO2, NO3 = Norge

FI = Finland

SE = Sverige

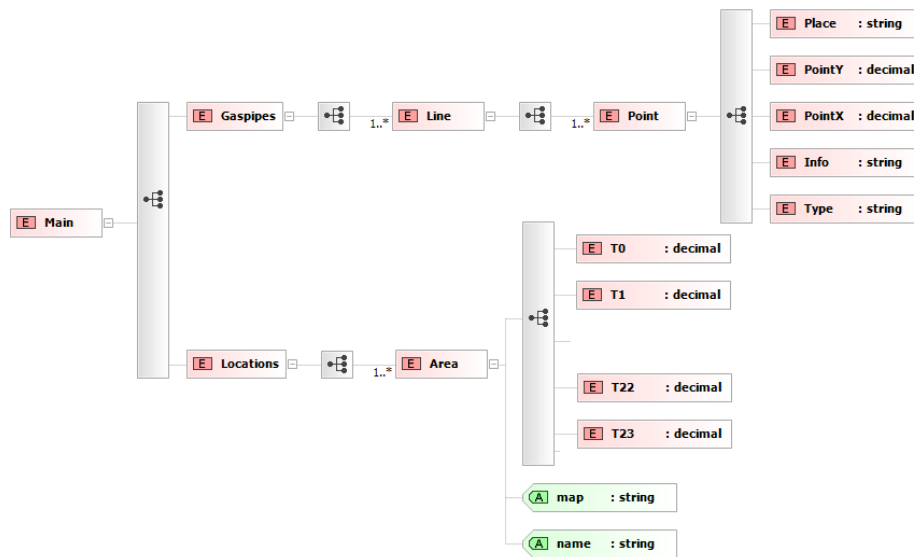
Vi kan se i denne illustrasjonen at vi har prisinformasjon for alle prissoner for hele døgnet time for time. Det er også dette som blir vist fram i Nord Pool Spots illustrasjonen som vi har brukt som modell. Ved å hente data fra Nord Pool Spot får vi realistiske priser som gjør det enklere for oss å forstå hvordan prisnivået varierer i løpet av et døgn, og prisforskjeller mellom soner.

2.3.2 Lagring av testdata

I vårt forslag til løsning bruker vi denne informasjonen for å illustrere priser for de ulike prissonene på ulike tider på døgnet. Prisene oppdateres kontinuerlig. Dette har Navita tatt høyde for ved å lage en kobling mellom sine løsninger opp mot en ekstern tjeneste. Siden Navita allerede har en løsning for dette har vi valgt å bruke statiske test data i våre løsninger. For å kunne fokusere på visualiseringsproblematikken og teknologiske plattformer ble vi enige med Navita om å ikke investere for mye tid i å bruke dynamiske priser.

Vi benyttet oss av XML formatet for å lagre testdata. I ett av løsningsforslagene (skrivebordsløsningen) kan vi velge mellom grafisk visualisering av prisinformasjon og tabellbasert visualisering av prisinformasjon. Her kan vi endre på prisinformasjonen, og oppdatert prisinformasjon vil være tilgjengelig både i grafisk og tabell visualisering. Vi har laget en skrivebordsløsning og en webløsning som begge benytter seg av samme testdata i samme XML-format.

Testdataene ble i utviklingsprosessen tilpasset hver løsning, dvs. at hver løsning benytter seg av en egen XML fil selv om de har mye til felles. I ettertid ser vi fordelen ved å kun benytte en enkel XML fil og fastsette denne ved bruk av XML-skjema. Dette vil definitivt være en fordel når man skal sette sammen de ulike løsningene til et komplett system, da man unngår å åpne flere ulike filer. XML-skjemaet som ble fastsatt er vist i figur 2.1 og de ulike elementene forklart under.



Figur 2.1: XML-skjema basis.

Main Rotelementet

Gaspipes Brukes for å visualisere rørledninger

Line Punktene inni hvert Line element blir knyttet sammen med rørledninger.

Point Innholder informasjon om hvert element(Oljeplattform, raffineringsanlegg osv.)

Place Navnet på lokasjonen.

PointY Breddegraden for lokasjonen i desimaltall.

PointX Lengdegraden for lokasjonen i desimaltall.

Info Informasjon tilknyttet punktet.

Type Type element (Land, Sea, Bend) blir knyttet opp mot valg av ikon.

Locations Brukes for å visualisere prisnivå for ulike områder.

Area Area er et område som skal være med i visualiseringen.

T0,T1..T23 Disse inneholder prisinformasjonen for et døgn for gjeldende område.

map map benyttes hvis området skal visualiseres ved bruk av et forhåndslaget overlay(gjelder webløsningen).

name ved å benytte name sendes denne variabelen til Geonames for å hente geografiske data om lokasjonen, for så å visualisere området ved bruk av sirkler.

Utdrag fra XMLfilen

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<Main>
  <Gaspipes>
    <Line>
      <Point>
        <Place>Troll</Place>
        <PointY>5.009168</PointY>
        <PointX>57.3780391</PointX>
        <Info>Oildrilling</Info>
        <Type>Sea</Type>
      </Point>
      <Point>
        <Place>Stavanger</Place>
        <PointY>5.573609000000008</PointY>
        <PointX>58.93352499999998</PointX>
        <Info>Oil refinery</Info>
        <Type>Land</Type>
      </Point>
      ...
    </Gaspipes>
    <Locations>
      <Area map="N01">
        <T0>13.64</T0>
        <T1>10.29</T1>
        ...
      </Area>
      <Area map="Sweden">
        <T0>29.75</T0>
        <T1>22.73</T1>
        ...
      </Area>
    </Locations>
  </Main>
```

2.4 Applikasjonstype

2.4.1 Skrivebordsløsning

I den opprinnelige oppgaveteksten fra Navita har de lagt opp til at oppgaven skal løses som en skrivebordsløsning. Navita har et godt samarbeid med Microsoft, derfor vil de ha laget dette i Microsofts utviklingsmiljø Visual Studio basert på C# med tilleggspakken Chart FX Maps.

Etter Navitas krav laget vi et løsningsforslag i Visual Studio. Underveis i utviklingsprosessen fant vi problemer med utviklingsverktøyets fleksibilitet for

å kunne løse utfordringene i oppgaven. Det var en utfordring å få til den interaktiviteten vi ønsket, og mener at kombinasjonen av utviklingsverktøy kunne være en av faktorene til dette.

2.4.2 Webløsning

En mulig løsning for dette er å skifte utviklingsverktøy. Navita ønsket i sin opprinnelige oppgavetekst å undersøke muligheter for å løse de samme problemene i internettbaserte kartprogrammer som Google Earth eller Microsoft Live Earth. Vi så på muligheter innenfor disse forslagene, og var litt usikre på om dette var et godt valg. Etter litt forskning på hvilke muligheter vi hadde, kom vi fram med ideen om å benytte Google Maps. Likhetene med Google Earth er mange, en av forskjellene er at Google Maps er basert for nettleseren.

Google Maps har gitt ut en API som er tilgjengelig for alle. Med denne APIen kan man bruke kartgrunlaget i Google Maps som basis i applikasjoner. Denne APIen benytter seg av JavaScript og mulighetene for legge til funksjonalitet på Googles kartverk er mange. Det finnes mange prosjekter som benytter seg av Google Maps kartgrunnlag for å visualisere sine data.

Kapittel 3

Bakgrunn

3.1 Børsbasert Krafthandel

Børsbasert krafthandel i Skandinavia, og spesielt Norge skilte seg ut fra resten av verden når det gjelder salg og prissetting av energi. Norge var det første landet i Skandinavia som privatiserte kjøp og salg av energi (strøm). Før denne privatiseringen i 1991 var det Stortinget som regulerte prisene på kjøp og salg av strøm. Etter privatiseringen er kraftprisene styrt av børsprinsippet, der medlemmene av børsen, enten produsenter eller kjøpere av energi kan selge eller kjøpe energi til priser som stadig forandrer seg [10]. Det er mange faktorer som avgjør prisen for kraft, men det tradisjonelle prinsippet om tilgang og behov er selvfølgelig en stor faktor når prisen fastsettes. Det er store geografiske områder med i bildet når hele Norden er med i denne kraftbørsen. Derfor er Norden delt inn i regioner, med forskjellige priser. Sverige, Danmark og Finland har sin egen pris region. I Norge er det litt mer komplisert. Hver uke blir regionsprisen oppdatert der kraftbørsen setter grensene for hvilke prissoner som gjelder for neste uke. Dette er et av problemene vi skal se på i vår oppgave, med dynamisk inndeling av kraftsoner. Dette vil forklares dypere når oppgaven presenteres senere i denne masteroppgaven.

Skandinavia er verdens første multinasjonale børs for handel av kraft. Noen av argumentene for å bruke en kraftbørs for at alle skal få like forutsetninger for kjøp og salg av kraft blir gjengitt nedenfor. Vi gjør oppmerksom på at disse argumentene ikke er våre argumenter, og at virkeligheten for enhver forbruker kan virke/er annerledes enn hva interessentene bak og de som jobber med kraftbørsen mener. Deres argumenter for å benytte oss av børsbasert krafthandel er som følger: Forbrukere kommer godt ut av dette på grunn av prispress fra resten av Norden. Norge kommer spesielt godt ut av internasjonal handel av energi. Dette er på grunn av Norges evne til å "lagre" strøm. I vanlige tilfeller kan ikke ferdigprodusert kraft lagres, denne kraften kjøres ut på linjene for å dekke etterspørselen av kraft. I Norge baserer vi mesteparten av kraftproduksjon på vannkraft, som har muligheter for å justere hvor mye kraft som skal produseres

ved å justere vannmengden som kjøres inn i generatorene. Når vi ser på de andre landene i Norden og Europa så brukes andre former for kraftproduksjon. Kraftverk basert på fossilt brennstoff er mye brukt i tillegg til atomkraft anlegg. Atomkraftverk er lite justerbare og produserer en lik mengde kraft døgnet rundt. Dette er svært fordelaktig for oss nordmenn som kan skru av vannkraftverkene på kvelden og importere billig atomkraft fra andre land. Importprisen blir lav fordi tilgang av energi er høy i forhold til etterspørsel. Når atomkraft nasjonene kommer inn i “peak perioden”, da industrien går for fullt og etterspørsel etter kraft er høy og spot prisene er høye kan vi i Norge eksportere dyr strøm.

3.2 Litt historikk om Nord Pool

Den tidligere omtalte nordiske kraftbørsen heter “NordPool”. Kraftbørsen ble etablert i 1993 og eid av Norske Statnett SF (50%) og Affärsverket Svenska Kraftnät (50%). Dagens NordPool generer 70 arbeidsplasser i Norden, samt Nederland og Tyskland. I det første driftsåret ble det handlet kraft for NOK 1.5 mrd (18.4 TWh). Fram til 1996 var NordPool kun for krafthandel i Norge. I 1996 ble det åpnet for krafthandel med Sverige. I 1998 kommer Finland med i NordPool og i år 2000 ble Danmark med i den nå nordiske kraftbørsen. Samme år blir det opprettet den første kraftbørsen i Tyskland (Leipzig Power Exchange) som NordPool har eierinteresser med 35%. På bakgrunn av dette får NordPool lov til å handle kraft med Tysklands to kraftbørser som kraftmegler. I 2005 utvider NordPool sitt produktspekter med å tilby kjøp og salg av EUs CO₂ kvoter. [10]

NordPool [10] er indirekte relatert til vår oppgave gjennom Navita som er vår oppdragsgiver. Navita er et selskap som utvikler løsninger for krafthandel. NordPool bruker Navitas produkter som kjerne i hverdagen med handel av kraft.

3.3 Litt historikk om Navita

Navita er en teknologibedrift som har sine røtter i Institutt for Energiteknikk (IFE). IFE fikk i 1988 et forskningsoppdrag som utgjorde startgruppen for verdens første kraftbørs, NordPool. El-Skandinavia ble dannet i 1996 og fokuserte på å lage systemer for krafthandel. El-Skandinavia ble kjøpt opp av OM Gruppen. Navitas ledelse kjøpte seg selv ut av OM Gruppen og dannet firmaet Navita i 2003.

I dag leverer Navita løsninger til selskaper som driver med kjøp og salg av såkalte “comodities”. Med andre ord så leverer Navita, gjennom hovedproduktet deres “PomaxTM” et verktøy for trading av elektrisk kraft, gass, råolje, kull, CO₂ utslipp, transport osv. På verdensbasis finnes det over 100 installasjoner av “PomaxTM” produktet. Navita har hovedkontor i Halden, og har flere underavdelinger. Store norske bedrifter som Norsk Hydro og Statkraft er på kundelisten til Navita. [3]



Figur 3.1: Nordpool spot illustrasjon

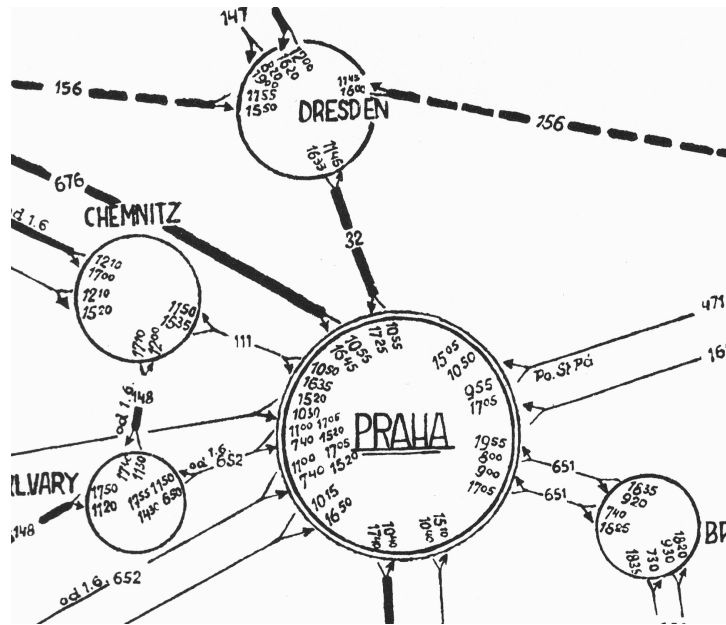
3.4 Visualiseringsteori

Det finnes forskjellige måter å visualisere gassnettverket på. Man kan benytte seg av et kartgrunnlag for å oppnå geografisk tilhørighet, eller man kan ta vekk kartgrunnlaget for å øke fokus på informasjonen man ønsker å formidle. Det finnes forskjellige visualiseringer av transportsystemer som er velegnet for bruk til gassnettverket. For å visualisere prissoner kan man f. eks. fargelegge hele sonen eller man kan fargelegge deler av sonen. De forskjellige måtene å markere prissoner på har sine fordeler og ulemper. Vi skal se på noen forskjellige måter å gjøre dette på og deres bruksområder.

3.4.1 Transportplan for fly

Denne modellen er basert på å ikke bruke et kartgrunnlag, hensikten er å fokusere på struktur og lesbarhet. Modellen er ikke geografisk korrekt, men har de geografiske lokasjonene visualisert i relativt forhold til hverandre. Modellen

er en ruteplan for Czechoslovakia Air Transport Company i 1933 og modellen baserer seg på sirkler og tilkoblinger mellom disse. Informasjonen som er viktig for brukerne er tidspunktene for når flyene til ønsket destinasjon tar av. Denne informasjonen kan med letthet presenteres i en rutetabell, men rutetabellen har ikke en grafisk oversikt over hvilke destinasjoner som er tilgjengelig. Ved hjelp av dette rutekartet/planen er det lett å finne ut hvilke destinasjoner som er tilgjengelig til og fra alle destinasjoner som er inkludert i nettverket. Modellen har visualisert et komplekst nettverk av flyruter med avgangstider og identifikasjonsnummer til et kart/ruteplan. [31]

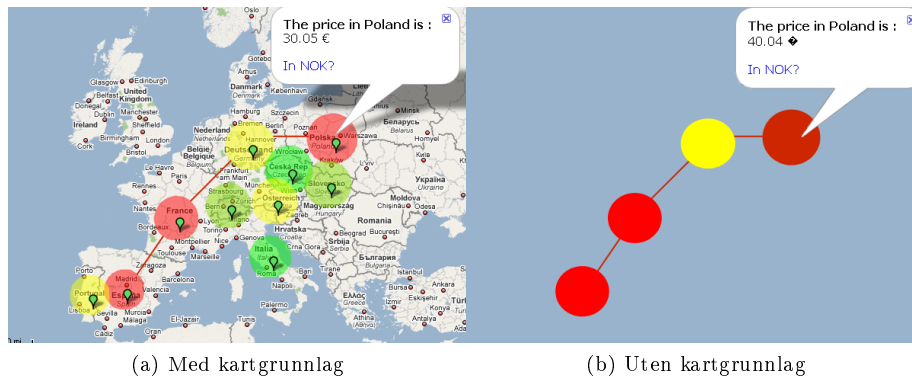


Figur 3.2: Ruteplan for Czechoslovakia Air Transport Company 1933. Bildet er hentet fra Envisioning Information av Edward Tufte[31]

Fordeler som vi kan se med denne løsningen er god lesbarhet over et relativt komplekst nettverk av flyavganger. Gassnettverket kan sammenlignes med nettverket som her er representert, ved at det har mange rørledninger, knutepunkter og noder.

Denne løsningen kan bli brukt for å visualisere gassnettverket med små modifikasjoner. Vi har laget et forslag som baserer seg på denne løsningen, ved at vi benytter sirkler for å markere prissoner. Vi har senere kalt denne løsningen for generelle prissoner (Se figur 3.3a). Denne løsningen er ganske generell så det vil være en triviell jobb å slå sammen vår løsning av generelle prissoner med figur 3.2 for å illustrere gassnettverkets oppbygging, både med og uten kartgrunnlag.

Informasjon som flyavganger og identifikasjonsnummer kan gjengis i dynamiske pop-up vinduer i tooltips. Dette er en bedre løsning da informasjonen ikke er synlig, men informasjonen er tilgjengelig ved at brukeren samhandler



Figur 3.3: Modifisering av generelle prissoner løsningen

med løsningen. Dette gjør at enda mer informasjon kan bli gjort tilgjengelig for brukeren. Denne dynamikken er ikke mulig å få til i papirbaserte løsninger, da må informasjonen legges til som vist i figur 3.2.

3.4.2 Visualisering av flyt

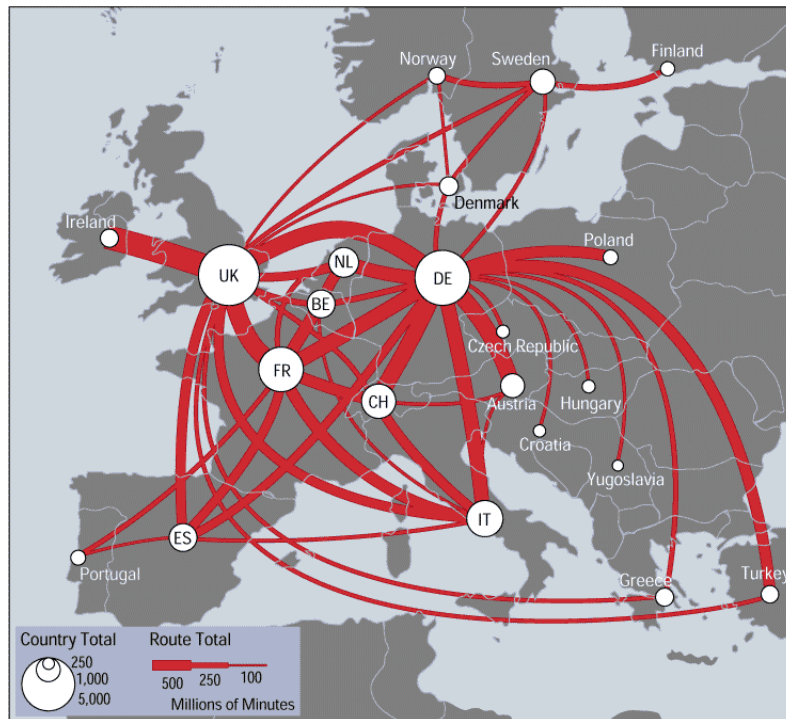
En del av våre arbeidsoppgaver omhandler gasslogistikk. En viktig del av gasslogistikk er flyt gjennom rørledninger. Det er flere måter å visualisere flyt gjennom rørledninger. En av måtene er bruk av farger som vi har beskrevet i seksjon 3.4.5. I denne seksjonen skal vi se på en litt annen løsning enn de løsningene som blir omtalt i dette kapittelet. Dette løsningsforslaget er tilpasset flyt mellom soner, i dette tilfellet land. Forslaget som blir beskrevet nedenfor viser flyt på en utmerket måte, men vi er skeptiske til at denne måten å løse visualisering av flyt på vil fungere i alle tilfeller. Forslaget baserer seg utelukkende på flyt, så det kan tenkes at dette forslaget er komplisert å kombinere med de løsningene vi har kommet frem til. Kartet er laget av Telegeography[12] som driver med forskning og analyser i telekommunikasjonsbransjen.

Figur 3.4 visualiserer telefonflyt mellom europeiske land. Tykkelsen på linjene mellom landene viser graden av flyt, da tykkelsen er linket opp til hvor mange minutter av telekommunikasjon som har gått mellom disse landene. Dette kan relateres til gassflyt ved å endre tykkelsen på rørledninger basert på hvor bra flyten er i rørledningene. Eksempelet gir ikke en eksakt verdi, men er ment som et oversiktsbilde som viser situasjonen i Europa. I eksempelet er det benyttet buede linjer for å øke lesbarheten. Det er flere interessante muligheter i dette kartet, både bruken av forskjellig tykkelse på rørledninger, men også de buede linjene som ikke er geografisk korrekte, men som illustrerer flyten mellom soner med god lesbarhet.

Vi mener at denne ideen kombinert med kartgrunnlaget fra Google Maps vil være en god løsning i fremtiden. Fordelene med denne løsningen vil være geografisk korrekt koordinatsystem, muligheter for dynamisk oppdatering av

datagrunnlag, og Google Maps interaksjonsegenskaper til brukeren.

Martin Dodge, forsker ved Centre for Advanced Spatial Analysis at University College London hevder i Mappa Mundi Magazine [13] at Telegeography har laget det beste flytkartet av nettbasert trafikk så langt.



Figur 3.4: Trafikkflyt kart for telekommunikasjon. Kartet er hentet fra Mappa Mundi [13]

3.4.3 Metrokart, London Underground

I oppgaveteksten fra Navita har de bedt oss om å undersøke muligheten for såkalt “smart-mapping”, eller en annerledes måte å tegne opp gassnettverket på. Som referanse har Navita oppgitt London Underground som eksempel. Denne filosofien er benyttet som en mal blant mange av dagens transportkart, selv om visse modifikasjoner har blitt gjort. Bakgrunnen for at denne fremvisningsformen ble laget var et visualiseringsproblem i datidens London Underground tidlig på 1900 tallet.

De fleste av verdens undergrunnsbaner har kart som er oversiktlig, men ikke alltid geografisk korrekte. Hensikten med disse kartene er at alle skal kunne sette seg inn i disse kartene rimelig raskt, slik at de vet hvor de skal bytte tog og lignende. Vi har tatt utgangspunkt i undergrunnsbanen i London, da dette

er den originale versjonen av metrokart, og prinsippene gjelder for de fleste undergrunnsbaner som baserer seg på Londons originale versjon.

Mannen bak prosjektet het Harry Beck og tegnet elektriske koblingsskjemaer for undergrunnsbanen i London. Det er hans bakgrunn fra dette fagfeltet som gjenspeiler seg i hans karakteristiske fremstilling av London Underground. Elektriske koblingsskjemaer blir laget med tanke på å være oversiktlige, noe som Harry Beck overførte til hans kartvisualisering. Kartet hans er ikke geografisk korrekt, men er til gjengjeld meget oversiktlig og lettlest. Ved bruk av fargekoder klarer Beck å skille de ulike rutene fra hverandre, noe Edward Tufte i tillegg kommenterer nedenfor. Harry Beck benyttet seg kun av horisontale og vertikale linjer, men har mulighet for 45graders bøy i kartet. Det første offisielle kartet som Harry Beck tegnet var i 1933. Dagens kart følger Becks prinsipper, selv om kartet har blitt oppdatert mange ganger.

Vurderinger

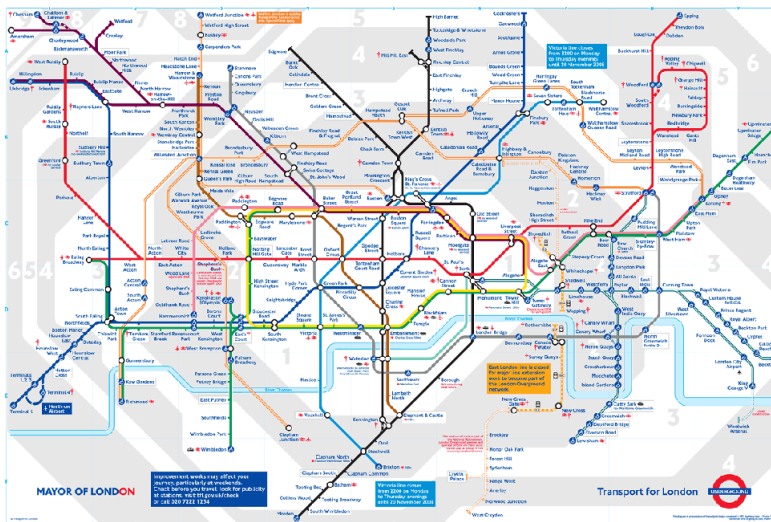
Edward Tufte har en diskusjon på sin egen hjemmeside der han kommenterer kartet til Harry Beck. Tufte skriver at Beck har samlet informasjonen på en vakker måte. Etter 70års revisjoner overlever fortsatt kartet. Videre skriver Tufte at dette kartet, er meget bra til sitt bruk. Historien gir han rett i dette, da mange av andre byers undergrunnsbaner baserer seg på denne kartfilosofien i større eller mindre grad. Denne måten å visualisere kart på er bra for dette formålet, men det er en mindre egnet arkitektur for generell visualisering av kart. Dette viser seg når vi skal forsøke å visualisere våre problemer ved hjelp av Becks premisser for å lage oversiktlig kart, men vi har laget noen eksempler som kan fungere. Tufte påpeker også at til tross for mange forsøk på å endre fremstilling, har de ikke lyktes med å finne noe som visualiserer denne informasjonen på en bedre måte. [29]

Tufte skriver at denne formen for visualisering er bra til sitt bruk, men en mindre egnet arkitektur for visualisering av annen type informasjon. På tross av dette har det blitt funnet andre bruksområder for bruk av London Underground metaforen. Keith V. Nesbitt[25] har forsøkt å visualisere abstrakt informasjon ved bruk av London Underground metaforen. Resultatet er interessant, og viser at det er mulig for enkelte typer informasjon og få en bra lesbarhet på informasjon som kanskje ikke er trivielt å visualisere på andre måter. Informasjon som har blitt visualisert med denne metaforen er brainstorming for en doktoravhandling, forretningsplan, sitemap for websider og kursoversikt ved et universitet. Dette viser at det fins potensial i denne metodikken.

Som vi ser av illustrasjonene, er det første offisielle kartet til Beck meget lik nåværende offisielle kart for London Underground. Dette er et bevis på kartets styrke, man har forsøkt å finne erstatninger til Harry Becks kart, men faller tilbake på hans premisser.



Figur 3.5: London Underground kart anno 1933



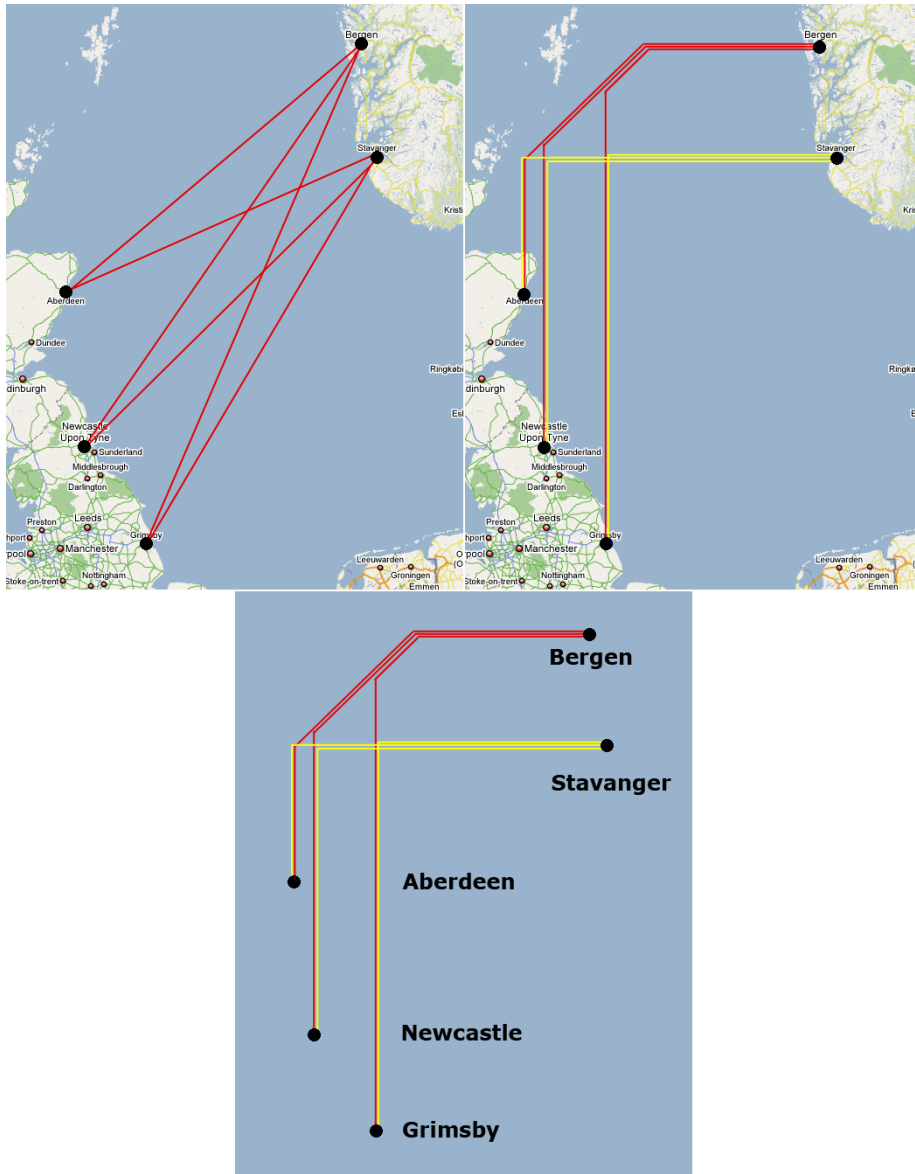
Figur 3.6: London Underground kart anno 2008

Varianter av metrokart filosofien

Vi har laget noen illustrasjoner som er forslag til ulike måter å benytte oss av London Underground metodikken. (Rørledninger som er brukt i disse illustrasjonene er ment for demonstrasjonsformål, derfor kan ledningenes koblingspunkter være feilaktige.) De tre ulike illustrasjonene bruker de samme rørledningene som strekker seg fra Bergen og Stavanger til Aberdeen, Newcastle og Grimsby i Storbritannia.

Forklaring av figur 3.7:

Bildet øverst til venstre viser tre rørledninger tegnet uten system på Google Maps kartgrunnlag. Neste bildet viser de samme rørledningene, men her har vi benyttet London Undergrounds premisser for å øke lesbarheten. Visualiseringen er fortsatt geografisk korrekt. Det siste bildet er tilsvarende bilde 2, men her har vi tatt vekk kartgrunnlaget for å øke lesbarheten ytterligere.



Figur 3.7: Ulike varianter av metrokart

3.4.4 Geografisk korrekte løsninger

En geografisk korrekt løsning innebærer at vi benytter et kartgrunnlag som er geografisk korrekt. Lager man en applikasjon som er avhengig av å være geografisk korrekt har man problemer dersom det viser seg at kartgrunnlaget ikke er korrekt. Starter man med feil i en av de viktigste byggeklossene, vil man

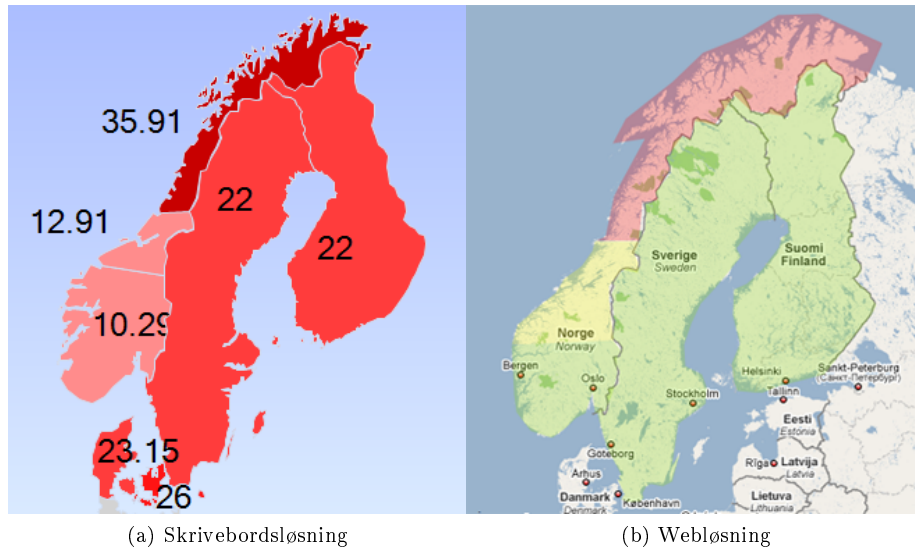
få feil i resten av applikasjonen. Hvordan kan man sikre seg at et kartgrunnlag er korrekt?

Mye av denne jobben ligger i planleggingen. Man er nødt til å finne en tjeneste eller leverandør som kan dokumentere at kartene deres er korrekte i henhold til en akseptert standard. I vårt tilfelle med skrivebordsløsningen brukte vi en teknologi som ikke er ment å være geografisk korrekt, eller har kart som er geografiske korrekte. Det viste seg at kartene vi benyttet var kjøpt fra tredjepart, og vi klarte ikke finne ut hvilken kartprojeksjon disse kartene var kodet i.

Vi mener at i en geografisk korrekt applikasjon må vi ha et koordinatsystem som alle elementer bruker. Det har ingen hensikt dersom kartgrunnlaget baserer seg på et koordinatsystem og elementene som blir lagt til i ettertid benytter seg av et annet koordinatsystem. For å sikre oss at løsningen er geografisk korrekt bestemte vi oss for å basere alle lokasjoner på lengde- og breddekoordinater. GPS systemer benytter seg av disse koordinatene, det samme gjør internettbaserte kartsystemer som Google Maps og Microsoft Live Earth.

Vi gjorde et forsøk på gjøre vårt eksisterende kartgrunnlag geografisk korrekt ved å konvertere kartets koordinater til lengde- og breddegrader. Dette er en operasjon som vi har beskrevet i seksjon 4.2.2. I praksis viste det seg å ikke være effektivt.

Vi har laget forskjellige løsninger som er geografiske korrekte. Vi har tidligere nevnt at vi hadde problemer med kartgrunnlaget på skrivebordsløsningen, men ser vi bort fra kartets koordinater har vi kommet fram til en løsning som er oversiktlig og delvis korrekt. Vi har ikke lagt til noen elementer som krever geografisk korrekthet. Dette betyr at informasjon som er visualisert i denne løsningen er korrekte i forhold til hva man ser på kartet, men det er ikke mulighet for å eksportere eller importere informasjon med denne løsningen. Man kan risikere at elementer havner på andre steder enn det man trodde på forhånd. Vi henviser til figur 4.7 hvor vi demonstrerer hvor avhengig man er av korrekte kartgrunnlag.



Figur 3.8: Forsøk på geografisk korrekte løsninger

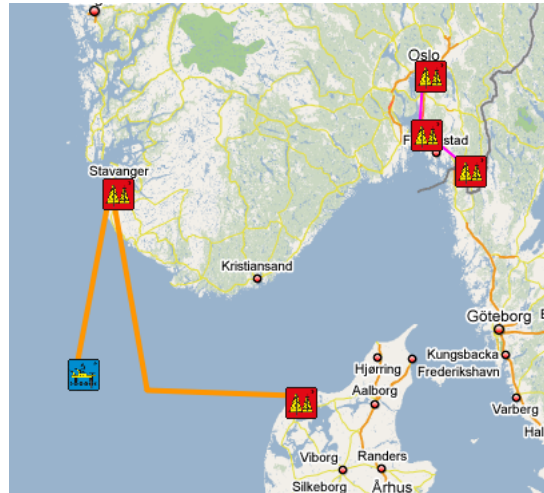
Fordelen med løsningen(3.8a) er at den er oversiktlig og lesbar. Informasjonen som er knyttet til dette kartet er prisnivået på elektrisitet. Prisnivået er representert ved tall og ved farge på prissonen. En lysrød farge vil indikere at prisen er lav, og en mørkrød farge tilsier at kraftprisen er høy. Det er ingen elementer her som er avhengige av korrekt geografisk lokasjon. Landene som er representert her er korrekte i forhold til hverandre, men dersom nye elementer skulle blitt lagt til ville vi måtte korrigere posisjonen, enten interaktivt eller med en matematisk interpolasjon. Dette fordi vi ikke vet lokasjon på byer, fylkesgrenser etc. Den eneste informasjonen vi kan forholde oss til er landegrensene.

Den forrige løsningen(3.8a) er ganske usikker med tanke på hvor man skal plassere nye elementer. Derfor bestemte vi oss for å teste en løsning hvor vi har et kartgrunnlag som vi vet er geografisk konsistent. Tankegangen var å benytte oss av Google Maps, for så å legge til den informasjonen vi ønsket som et nytt lag på kartgrunnlaget. Denne filosofien fungerte bra, og med hjelp av en bra API klarte vi å løse flere utfordringer hvis vi sammenligner med skrivebordsløsningen.

Figur 3.8b er en webløsning hvor vi har benyttet kartgrunnlaget til Google Maps. Med dette kartgrunnlaget er vi sikre på at vi har et geografisk korrekt kartgrunnlag. Denne illustrasjonen viser et eksempel som er tilsvarende skrivebordsløsningen i figur 3.8a. Her har vi benyttet en annen måte å illustrere prisene på. Vi har gått bort i fra å sette prisene direkte på sonen, men heller plassert de i en liste under selve kartet. Dette gjorde vi for at Navita har muligheten til å velge hvilken måte de ønsker informasjonen presentert. I denne løsningen har vi benyttet en annerledes fargeskala. Her har vi benyttet fra grønt = billig og rødt = dyrt. Hvorfor dette har blitt gjort kommer vi tilbake til i avsnittet om

bruk av farger.

Illustrasjonen under viser et forslag på hvordan gassnettverket kan visualiseres geografisk korrekt. Alle punkter er referert med lengde- og breddegrader i datagrunnlaget, noe som betyr at alle elementer er dynamisk opptegnet. Løsningen leser all lokasjonsinformasjon fra datagrunnlaget og tegner opp dette kartet: 3.9



Figur 3.9: Forsøk på geografisk korrekt webløsning

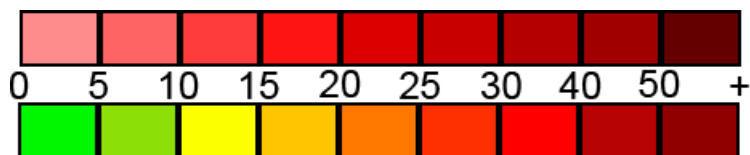
3.4.5 Bruk av farger

Det menneskelige øye har meget god sans for fargenyanser. Et trent øye kan skille mellom 1 million farger. For de fleste mennesker er over 20 000 farger tilgjengelig. På grunn av menneskets hukommelse, bør vi ikke bruke mer enn 20 til 30 ulike farger når vi skal fremstille abstrakt informasjon. [31] [18]

Vi har benyttet farger som verktøy i våre løsninger. Det er vanlig i geografisk sammenheng å benytte fargekoding for å illustrere f. eks høyde eller dybde. Vi har valgt kartet “General Bathymetric Chart of the Oceans” [26] som referanse. Dette kartet illustrerer havets dybde og fastlandets høyde ved hjelp av en skala med 20 farger. En mørkblå farge representerer større dybde enn en lys blå farge. På land har vi fra lys gul som er lavlandet, og mørk oransje som er de høyeste fjellene.

Vi har imidlertid brukt farger på forskjellige måter. På webløsningen har vi en skala som illustrerer kraftprisen. En skala med 9 punkter som går fra grønt til rødt, hvor rødt indikerer høyest pris, se illustrasjon 3.10.

På skrivebordsløsningen har vi også en skala på 9 punkter. Denne gangen går skalaen fra lys rød (billigst) til mørk rød (dyrest). Bakgrunnen for dette er mulig fargeblindhet hos brukeren. Leo M. Hurvich, 1981 skriver at man helst bør unngå å ta kritiske valg basert på at personen kan skille mellom grønn og



Figur 3.10: Illustrasjon av prisskala der vi ser prisskalaen til skrivebordsløsningen øverst og skalaen til webløsningen nederst. (Tallene referer til hvilken pris som gjelder til hver farge.)

rød farge.[18] Vi valgte å bruke grønn til rød skala på webapplikasjonen. Dette fordi vi er usikre på om fargekodingen i prissonene er av kritisk art, og dels fordi det er veldig vanlig å benytte grønn = bra, og rød = dårlig/mindre bra. Ved å visualisere to forslag kan Navita selv bestemme hvilken pris skala som passer deres behov. [31]

Vi har benyttet farger for å illustrere prisnivået i de ulike prissonene, men det er flere elementer som med fordel kan fargekodes. Ved å fargelegge rørledninger korrekt kan vi tiltrekke brukerens oppmerksomhet dersom det er nødvendig. På hvilken måte bør vi benytte farger hensiktsmessig?

Det finnes flere måter som alle er “riktig” fargekoding av rørledninger. Vi kan f. eks fargelegge etter status på rørledninger. Med status mener vi operasjonell status. Rørledninger som ikke fungerer optimalt kan fargelegges gult for mindre avvik, eller rødt for stort avvik/brudd på kabel. En annen måte å fargekode rørledninger er med tanke på flyt. Dersom flyten er optimal vil grønn være en god farge. (En må ta standpunkt til om bra flyt er maksimum overføringskapasitet, eller at bra flyt betyr mye kapasitet til overs.) Middels god flyt kan representeres ved gul og rødt brukes for å representere dårlig flyt. I de tre foregående eksempler har vi benyttet gule og rød rørledninger. Dette fargevalget har ingenting med kapasitet eller flyt å gjøre, men er fargekodet på denne måten for å øke lesbarheten. Eksempelene er laget for illustrere visualiseringsmuligheter.

3.4.6 Detaljeringsgrad

Gjennom prosjekttiden hadde vi ikke tilgang på et fullstendig bilde av gassnettverkets kompleksitet. Derfor var vi avhengig av Navitas tilbakemeldinger og vurderinger var korrekte. Vi kan imidlertid forestille oss at enkelte rørledninger er viktigere enn andre. Dette kan sammenlignes med kart over veinettet. I et oversiktskart over Norge vil ikke alle veier være inkludert, til dette trengs det utsnitt av kartet (zoom). Det er hovedveier og de viktigste veiene som blir tatt med på oversiktskartet. Vi mener det er fornuftig å bruke denne modellen ved visualisering av gassnettverket i fremtiden. Dersom vi hadde vært i en situasjon hvor vi hadde hatt ett løsningsforslag å jobbe med, og ressurser til å implementere gassnettverket ville det vært naturlig å bruke veikart metodikken. Vi har imidlertid jobbet med en løsning som har blitt utviklet med tanke på detaljeringsgraden. Størrelsen på ikoner blir endret fra fullverdige ikoner til enkle prikker.

Det er en utfordring å visualisere små detaljer i en så stor omgivelse som den som har blitt brukt i dette prosjektet, der vi har benyttet oss av kartgrunnlag av deler av Europa. Joachim Böttger et. al skriver i sin publikasjon, Detail-In-Context Visualization for Satellite Imagery [16] *“The intention of showing small details on the Earth’s surface while preserving the overview of the whole world or its parts is a typical detail-in-context problem”*. [14] [19]

Vi kan forklare dette med at en plattform er av liten geografisk størrelse i forhold til f.eks. et land eller et kontinent. For å gjøre plattformen synlig må vi lage en markering i kartet som er synlig for brukeren. Dette ikonet må være ganske stort i forhold til faktisk størrelse, og med tett forekomst av interessepunkter kan man risikere kaos av markeringer som fører til dårlig lesbarhet.

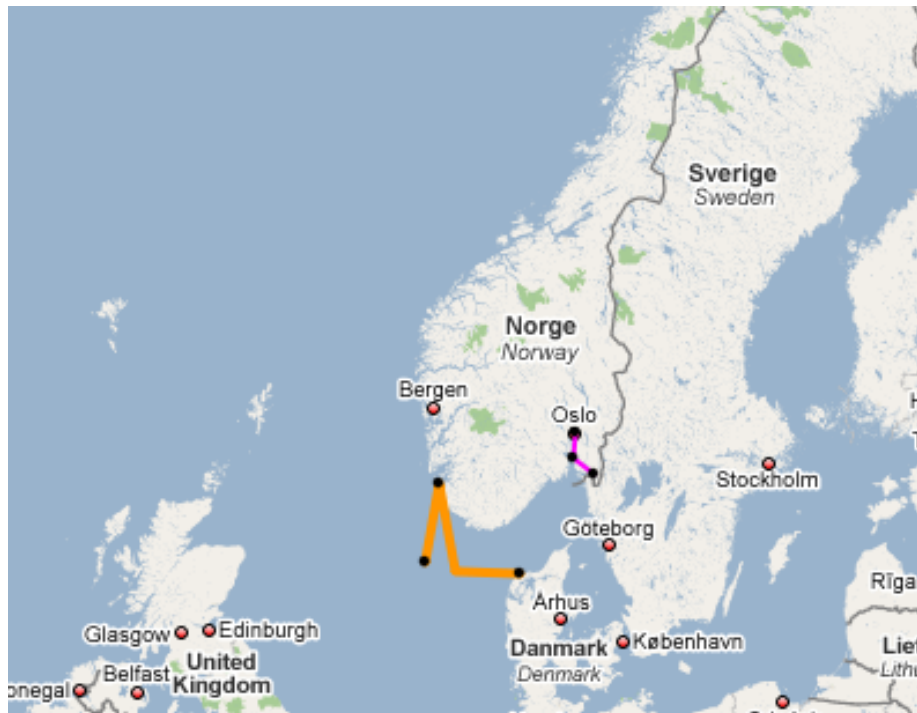
For å løse dette problemet er man nødt til å tenke på detaljeringsgraden. I våre løsningsforslag har vi ikke så høy tetthet av interessepunkter, men vi har likevel tenkt på detaljeringsgraden (detail-in-context) for å unngå problemer med lesbarhet. Vi henviser til figur 8.2 hvor flere av ikonene overlapper hverandre. Dersom det hadde vært enda flere hendelser ville lesbarheten blitt meget dårlig. Dette eksempelet burde tatt mer hensyn til detaljeringsgrad.

Vi har valgt å illustrere dette med ved å bruke ulike nivåer av zoom der vi viser forandring i ikon størrelse.

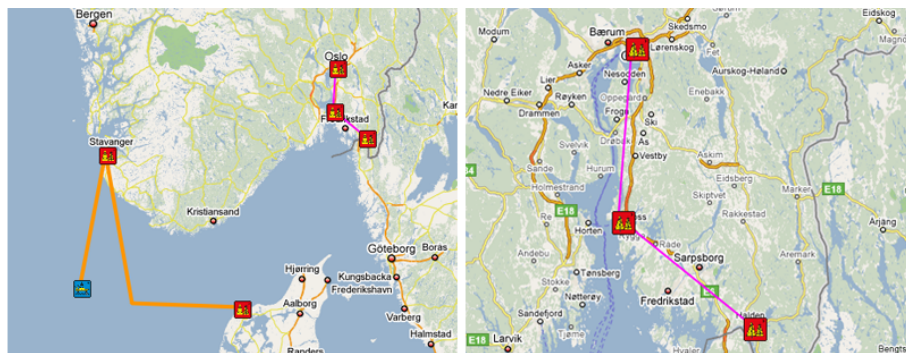
Google Earth har en interessant løsning på detaljeringsgradproblemet atikken med eksploderende ikoner. Med dette mener vi små ikoner som ikke nødvendigvis har en betydning, men oppfordrer bruker til å interagere med ikonet for nærmere informasjon.

Illustrasjon av detaljeringsgrad

Illustrasjonen inneholder tre bilder. Det store bildet, figur 3.11 viser hvordan interessepunkter blir fremstilt i et oversiktsbilde med et stort kartområde synlig. Her har vi benyttet oss av prikker for å fremstille et punkt. Vi mener at bruk av ikoner i denne visningen vil være u hensiktsmessig. En reell situasjon vil inneholde flere punkter enn vårt eksempel og da kan lesbarheten bli dårligere.



Figur 3.11: Kontinentalvisning



(a) Nasjonalvisning

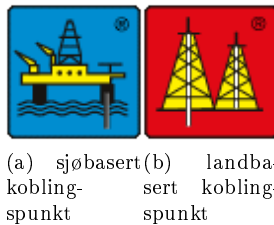
(b) Distrikvisning

Figur 3.12: Detaljeringsgrad

Figur 3.12 viser hvordan ikonene tilpasser seg til valgt nivå av zoom. Vi kan se at 3.12a har mindre ikoner enn 3.12b, som gjør at mulighetene er større for oppnå god lesbarhet ved senere utvidelser.

Ikoner

Vi bruker ikoner for å gi brukeren informasjon om hvilke type koblingspunkt som er benyttet. I figur 3.13 viser vi ikoner for sjø- og landbaserte installasjoner. Disse ikonene er laget for å benyttes under utviklingen av løsningene, på bakgrunn av at Navita ønsker å benytte egne ikoner er ikke disse ikonene nødvendigvis de best egnede. (Disse ikonene kan f. eks ha en litt for høy detaljeringsgrad) Dette er kun et forslag til Navita som kan utvide med så mange ikoner som ønskelig. Dette har blitt tatt høyde for under design og implementasjon av løsningene.



Figur 3.13: Ikoner

3.4.7 Dynamisk oppdatering

Gjennom utviklingsperioden har det hele tiden vært fokus på dynamisk opp-tegning av gassnettverket. Vi har designet løsningene på en slik måte at ingen elementer er statiske. Alle elementer blir tegnet på kartgrunnlaget dynamisk. Med dynamisk mener vi at all informasjon (foruten kartgrunnlaget) ligger lagret i XML-filer. Her vil koordinatene til en rørledning ligge med start- og slutt koordinater. Plattformen og koblingspunkter ligger lagret med x og y koordinater for plassering og har attributter som forklarer hvilken type punkt det er, og ekstra informasjon. Vi henviser til XML-skjema (seksjon 2.1) for fullstendig beskrivelse av XML-filene.

Grunnen til at dette blir gjort er at Navita skal ha mulighet til å oppdatere data som ligger i våre løsninger dynamisk. Målet er å benytte XML-skjemaet til Navita, slik at Navita kan kjøre oppdateringer i sin eksisterende løsning (PomaxTM), og sende de oppdaterte XML-filene til våre løsninger. På denne måten vil våre løsninger følge endringene til Navitas eksisterende system, og de kan til enhver tid være sikre på at all data er oppdatert.

Kapittel 4

Teknisk bakgrunn

Denne delen av bakgrunnskapittelet omhandler de ulike tekniske standarder som vi har jobbet med gjennom denne masteroppgaven. Basert på oppgavebeskrivelsen er det klart at vi har benyttet Visual Studio.net og C# samt tilleggspakken Chart FX Maps til å utvikle skrivebordsløsningen. Til å utvikle webløsningen har vi benyttet oss av flere teknologier som Google Maps og JavaScript, server delen er utviklet i aspx.net og C#. Alle teknologier og formater som vi har jobbet med er beskrevet i dette kapittelet.

4.1 Verktøy og standarder

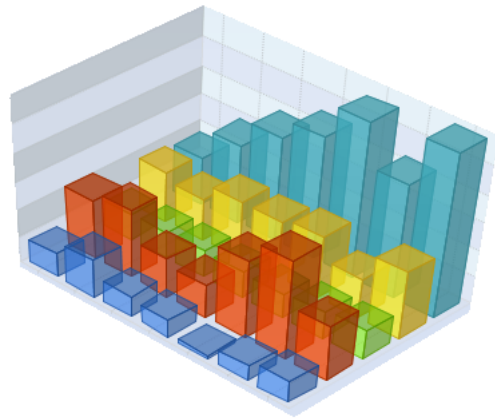
4.1.1 Visual Studio.net og Chart FX Maps

Visual Studio.net er Microsofts velkjente utviklingsmiljø for programmering for ulike Windows plattformer. Visual Studio serien ble introdusert i 1997, og hadde i hovedsak støtte for C++ 5.0 og Visual Basic 5.0. Ett år senere ble Visual Studio 6.0 lansert, og denne versjonen var gjeldene helt fram til 2002. I dette året ble .net rammeverket for Windows lansert. Filosofien bak .net rammeverket var økt kompatibilitet mellom forskjellige operativsystemer. En rekke biblioteker og klasser var bygget inn i dette rammeverket som utviklere kunne bruke for å lette sin jobb ved å bruke prefabrikerte kodesnutter. En fordel med denne filosofien gjør at applikasjoner utviklet med .net biblioteker skal kunne fungere på alle operativsystemer med korrekt .net versjon installert.

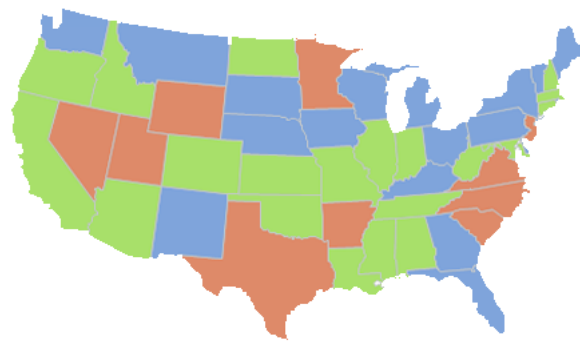
Ved å bruke .net rammeverket vil vår applikasjon være framtidsrettet da applikasjonen skal kunne kjøres på flere versjoner av Windows. Valg av programmeringsomgivelse er bare en del av ulike valg som må tas. Visual Studio har støtte for utvikling i flere programmeringsspråk, blant annet Visual Basic.net og C#. Med tanke på valg av programmeringsspråk hadde ikke Navita et spesielt ønske om hvilket språk som ble benyttet. De ulike valgene vi har innen Visual Studio pakken er C/C++, Visual Basic, C# og J#. Valget falt på C#. C# er fullt kompatibelt med .net rammeverket, objektorientert og et moderne språk med godt utbygde biblioteker. Navita var enige i vår argumentasjon

å benytte oss av C#.

Chart FX er et fremvisningsverktøy laget for Visual Studio av Software FX. [11] Mange selskaper benytter seg av Chart FXs API for å vise frem sine data. Av navnet så er det Charts, altså diagrammer som er Chart FXs sterkeste side. Vi skal vise frem kart, og ikke diagrammer. Til dette finnes tilleggspakken Maps, som er en samling med kart laget i SVG formatet. Etter lovende ord og referanser fra Software FXs hjemmeside støtter vi Navitas valg om å prøve å lage en løsning basert på disse verktøyene. Under utviklingstiden fant vi sterke og svake sider ved Chart FX. De sterke og svake sidene ved Chart FX vil bli beskrevet i detalj i konklusjonen og i kapitlett Teknisk beskrivelse.



Figur 4.1: Illustrasjon av Chart FX diagram fremvisning.



Figur 4.2: Illustrasjon av Chart FX Maps fremvisning.

4.1.2 Extensible Markup Language

Extensible Markup Language(XML) er laget for å strukturere eller beskrive data i elementer ved bruk av tagger. XML inneholder ingen predefinerte tagger, slik som for eksempel HTML, noe som gjør det meget allsidig. Det er opp til brukeren å definere de elementene som systemet krever. XML er laget for å lagre data, og inneholder ingen funksjonalitet for å vise det frem. Dette gjør at XML er avhengig av ekstern programvare for å sende, motta eller vise frem dataene. Da XML kun er vanlig tekst, er det fullstendig maskin- og programvare uavhengig, noe som gjør det mulig å implementere i all programvare som kan lese standard tekst dokumenter. I applikasjonene vi har utviklet ble det ikke stilt noen krav om lagringsmetoder, da Navita har sine egne rutiner for å håndtere dette, så vi valgte å benytte oss av XML. En av grunnene er at dataene kan benyttes i begge applikasjonene. En annen fordel er at det ikke krever noe oppsett, slik som SQL database, og da dette kun skulle benyttes til testdata ble dette en viktig faktor. Det er dessuten slik at en overføring av data mellom xml-baserte data og en eventuell database er svært enkelt og sikkert. Under følger et eksempel på hvordan vi har lagret en posisjon med navn, GPS-koordinater, info og type objekt. Denne posisjonen illustrerer Troll plattformen. (Lengde- og breddegrader er for demonstrasjons formål, og kan derfor være ukorrekt)

```
<Point>
  <Place>Troll</Place>
  <PointY>5.009168</PointY>
  <PointX>57.3780391</PointX>
  <Info>01jeplattformen Troll</Info>
  <Type>Sea</Type>
</Point>
```

Vi fant det også hensiktsmessig å benytte oss av XML da vi i følge oppgavebeskrivelsen må benytte oss av SVG i en del av oppgaven. [34]

4.1.3 XML-skjema

XML skjema benyttes for å beskrive oppbygningen av en XML fil og blir sett på som en erstatning for Document Type Definition(DTD). Et skjema gir mulighet for å validere XML filer med hensyn til struktur og innholdsformater. Hovedforskjellene mellom DTD og skjema er at skjema er utvidbart for fremtidige tillegg, er skrevet i XML og inkluderer data typer og navneområder. Skjemaet gir mulighet for å definere:

- elementer
- attributter
- antall elementer og rekkefølgen på disse

- data typer og standard verdier for elementer og attributter

En instans av XML skjema kalles XML Schema Definition(XSD). Vårt XML-skjema er beskrevet i seksjon 2.1

4.1.4 Scalable Vector Graphics

Chart FX Maps benytter seg av vektorbasert grafikk på sine kart. Fordelen med dette er en meget god skalerbarhet som man ikke får til ved bruk av punktgrafikk. Med denne skalerbarheten kan vi benytte oss av zooming for bedre detaljvisning og lesbarhet av mindre områder. SVG er definert av W3C.

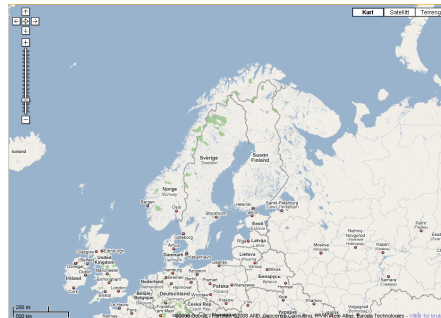
SVG er et XML format, da data lagres i en XML fil, og en programmerbar API for grafiske applikasjoner. I denne oppgaven er vi ikke inne på den programmerbare delen av SVG, da vi ønsker å programmere funksjonaliteten i Visual Studio og Google Maps API. En editering av SVG fil kan derfor gjøres på samme måte som man editere en XML fil, eller man kan bruke en SVG editor (grafisk) som f. eks Adobe Illustrator. Det er mange store aktører som støtter dette formatet, blant dem er Adobe, Microsoft, IBM med flere. [20]

Fordeler vi har ved å benytte SVG filer som kartgrunnlag er mer enn bare skalerbarhet. Filene er editerbare på samme måte som XML dokumenter, og er derfor lettoversiktelige.

4.1.5 Google Maps

Den andre delen av oppgaven som omhandler løsningen for web baserer seg på bruk av Google Maps. Google Maps er en karttjeneste som er laget av Google. Denne tjenesten inneholder kart over hele verden, og benytter lengde- og breddegrader som koordinatsystem. Google Maps støtter zooming tilstrekkelig for vårt bruksområde. Kartgrunnlaget ligger på Googles server. Dette betyr at kartgrunnlaget blir oppdatert regelmessig, og det eneste brukerne trenger å tenke på er å ha en internetttilkobling. Det viser seg å overføre kartdata i sanntid fra Googles servere er uproblematisk, Google har fått til et bra konsept på overføring av relativt store data på kort tid. Som bruker med en rimelig internett tilkobling vil en knapt merke ventetiden fra man sender en forespørsel til kartet er oppdatert, enten vi snakker om endring av zoomnivå eller forflytning i kartet.

Mulighetene for å legge til informasjon som er relevant for oss er store. Det finnes en tilgjengelig API som gjør at man med JavaScript kan programmere



Figur 4.3: Illustrasjon av Google Maps

mot sin egen versjon av Google Maps. Her kan man konfigurere og legge til objekter. Det finnes mange prosjekter som bruker Google Maps API, dette var en av grunnene til at vi valgte å teste ut Google Maps i dette prosjektet. Dokumentasjonen er god, og det finnes mange inspirasjonskilder [21] [4]. Andre masterstudenter benytter seg av Google Maps API i sine masterprosjekter. På bakgrunn av dette kan vi snakke med våre medstudenter for hjelp og innspill.

4.1.6 KML

KML filer brukes for å gjengi overlays i både Google Maps og GoogleEarth. Vi henviser til figur 4.5b for å se et overlay laget i Google Maps bli vist i GoogleEarth. KML er en åpen standard som blir vedlikeholdt av Open Geospatial Consortium og den baserer seg på XML. Strukturen for en KML fil er hierarkisk og er oppbygd av et sett med forhåndsdefinerte elementer. En KML fil kan beskrive ulike elementer, som f.eks. punkt, linje, polygon etc. Google Earth, Google Maps, NASA WorldWind, Adobe PhotoShop og AutoCAD har mulighet for å vise frem KML filer.

I prosjektet har vi benyttet oss av KML standarden for å manuelt opprette overlays i Google Maps. Når vi oppretter et polygon i Google Maps blir dette lagret i en KML fil for å ha mulighet til å vise denne som et overlay i Google Earth og ekstrahere polygon dataene til bruk i Google Maps. Det er mulig å vise frem KML filene direkte i Google Maps, men funksjonaliteten blir begrenset i forhold til å opprette polygonene ved bruk av Google Maps APIen.

4.1.7 Aspx.net

I forrige avsnitt snakket vi om overlays. Figuren som vi refererte til viser et manuelt laget polygon som skal forestille Østfold fylke. Muligheten for å kunne lage manuelle overlays er nyttig når ingen andre måter for oppdeling av prissoner er logisk. Her støtte vi på et problem som gjorde at vi måtte ha en mulighet for å lagre polygonene som vi tegnet manuelt på et Google Maps kartgrunnlag, for at det skal være mulig å bruke de på nytt. Tidligere nevnte vi at Google Maps API er beregnet på JavaScript. Problemer med å lagre til server som er problematisk, men ikke umulig i JavaScript gjorde at vi ønsket å benytte et annet språk for undersøke hvilke muligheter vi hadde med tanke på lagring til server. Valget falt på aspx.net.

Aspx.net er en videreutvikling av Microsofts asp (Active Server Page), som er Microsofts alternativ for å lage dynamiske websider. De eksisterende kodesnutter av JavaScript ble videreført, men det laget ekstra funksjonalitet som ble skrevet i asp.net. Den ekstra funksjonaliteten vi trengte var en lagringsfunksjon, som tok vare på inndelingen vi tegnet. Dette foregikk ved å legge til en lagreknapp på utsiden av kartbildet, som sendte polygoninformasjon til en webserver. Fra denne serveren kan vi hente opp lagrede polygoner. Ved ønsket tileggsfunksjonalitet i fremtiden er det en fordel at vi benyttet aspx.net. Da vi allerede har laget rammeverket kreves det relativt liten innsats for å være i stand til å legge til funksjonalitet utover JavaScripts begrensninger.

4.1.8 Geonames

Geonames er en geografisk database tilgjengelig for alle via internett. Databasen inneholder over 8 millioner geografiske navn, herav 6.5 millioner er unike geografiske lokasjoner. Informasjonen som ligger i denne databasen er brukergenerert, det vil si at brukerne av tjenesten legger inn og editerer de geografiske dataene selv. Geonames tilbyr en rekke tjenester for at alle skal kunne koble seg til databasen via en webservice. Det finnes flere måter å gjøre dette på, vi benyttet oss av JSON for å kommunisere med Geonames database. Mer om JSON i seksjon 4.1.9. Webservicene er gratis for alle. Geonames er populær og betjener opptil 11 millioner forespørsler hver dag.[7] Man kan stille spørsmål til om brukergenerert innhold er korrekt og sikkert nok. Folk kan legge inn feil informasjon, men med så mange forespørsler hver dag og brukere som NRK(www.yr.no), BBC, Nike, Adidas m. flere vil vi anta at informasjonen som ligger i Geonames database er av ganske god kvalitet.

4.1.9 JSON

JavaScript Object Notation er et tekst basert format for å utveksling av informasjon. JSON er mye brukt innen Ajax web applikasjoner, men siden det er et data format kan man benytte JSON i alle applikasjoner hvor man har behov for å lagre eller utveksle strukturert tekst informasjon. Enkelheten ved JSON er det som gjør det så nyttig, en JSON formatert melding består kun av et rot object eller array. Array elementene og objekt verdiene kan være objekter, arrayer, strenger, tall, boolske verdier og så videre. Men på en annen side mangler det viktige begreper som for eksempel mulighet for å lagre dato-og tidsvariabler, noe som blir benyttet mye innen data utveksling. Det er mange ulike data utvekslings formater å velge mellom, deriblant XML, men det er viktig å velge korrekt utvekslings format basert på hvilket behov applikasjonen har. XML har sine røtter i mark-up språket, mens JSON stammer fra ulike programmeringsspråk, noe som gjør JSON mer naturlig og lettere å utveksle strukturert informasjon. I prosjektet er JSON benyttet for å utveksle geografisk informasjon til og fra geonames. Valget til bruk av JSON falt på bakgrunn av at løsningen er basert på JavaScript, noe som da minsker behovet for ekstra parsing av informasjonen.[15]

4.2 Utfordringer og forslag til løsninger

4.2.1 Inndeling av prissoner

En av de viktigste funksjonene til løsningen Navita ønsker seg er inndeling av prissoner. Denne funksjonen var også noe av det første vi startet å implementere i de ulike løsningene. Vi refererer til illustrasjonen om Nord Pool Spot som viser hvordan det kan gjøres. Figur 3.1.

Datastruktur

En god måte å visualisere informasjon på kart er å visualisere strukturerte data. I teorien skal man kunne visualisere strukturerte data på hvilket som helst kartgrunnlag. Når det gjelder utviklingen av skrivebordsløsningen arbeidet vi med å strukturere data, noe SVG-formatet hjalp oss med.

Når det gjelder webløsningen hadde vi ikke en datastruktur som organiserte data. Angrepsvinkelen på denne oppgaven var derfor litt annerledes. Her ble det tatt snarveier i datastrukturingsfasen da det er tidkrevende å lage en tilsvarende datastruktur for webløsningen. Vi ser i ettertid at det hadde vært en fordel med en hierarkisk datastruktur, spesielt for soner med dynamiske inndelinger. På tross av den manglende datastrukturen har vi laget noen forslag til hvordan vi kan løse disse problemene uten å benytte oss av et strukturert datagrunnlag.

Skrivebordsløsning

Her benyttet vi oss av SVG filer som har en strukturering av data i grupper. Gruppene er påkrevd for at SVG filene kan brukes i Chart FX. Her vil et polygon, som f. eks inneholder omrisset av Sverige være lagret som Sverige. I ettertid vil det være en triviell jobb å markere polygonet Sverige med en farge som illustrerer prisnivået i denne sonen. Stort sett de fleste land har en egen prissone. Norge skiller seg ut ved å ha tre dynamiske prissoner, hvilket innebærer at det kreves en spesial løsning for å fremstille prissonene i korrekt.

Forslag til løsning for dynamiske prissoner

Navitas løsning har i dag en 150 soners inndeling av Norge. Denne løsningen er ikke 100% korrekt, da kompromisser har blitt inngått for å få en grei oppløsning. Etter møter med Navita fant vi ut at prisen er alltid den samme innenfor en kommune. Dermed kan vi basere oss på en oppløsning på kommunenivå er tilstrekkelig som gir oss en oppløsning på 430 soner. De ulike prissonene i Norge vil da bestå av et variabelt antall kommuner, som endres fra uke til uke. Setter man sammen de kommunene med den samme prisen vil man få en prissone. Et forslag til struktur kan være:

```
<Land>
  <Fylke>
    <Kommune>Polygon koordinater</Kommune>
  </Fylke>
</Land>
```

Vi har ikke har gjort denne jobben på grunn av manglende data om Norges kommuner. Dersom informasjon om kommunegrenser hadde vært tilgjengelig kunne vi tegnet opp et polygon per kommune. For å sette sammen en prissone bygd opp av kommunepolygoner, trenger man kun å vite hvilke kommuner som skal være med, for deretter å sette sammen en liste av polygoner som skal være aktive. Dette er et forslag som vil fungere på våre løsninger. Men selv om det

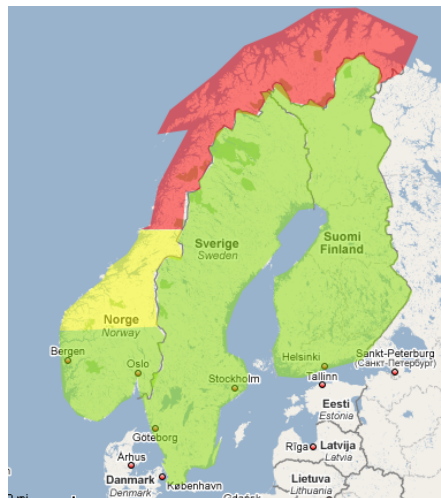
er mulig å få tilgang til geografisk informasjon om Norge vil det muligens by på større problemer i andre land. Løsningene er laget for å kunne dekke hele verden og tilgangen til regionale grenser vil være forskjellig fra land til land, noe som vil føre til en lite robust datastruktur.

Webløsning

I skrivebordsløsningen hadde vi tilgang til strukturerte data gjennom SVG kartene som fulgte med Chart FX Maps. For Google Maps måtte vi gjøre denne jobben selv. På grunn av manglende koordinatdata om land-, fylke- og kommunegrenser måtte inndelingen gjøres manuelt, noe som er tidkrevende og har derfor ikke blitt fullstendig implementert. Hvordan skal vi gå frem for å dele inn i prissoner på Google Maps kartgrunnlag?

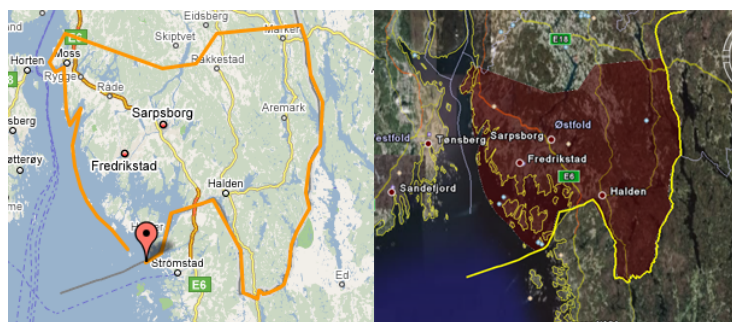
I ettertid innser vi at det hadde vært en fordel dersom vi hadde hatt muligheten til å lage hierarkisk strukturerte data. Dersom vi i tillegg hadde fått tilgang til kommunegrensene i Norge, hadde mulighetene for å få en 100% korrekt inndeling av Norge og resten av Europa vært store. Det er likevel mulig å lage soneinndelinger i Google Maps. Vi har laget noen demonstrasjonsløsninger som vi har vist Navita. Disse demoene har som hensikt å illustrere hvilke muligheter det ligger i teknologien.

Overlay Vi har laget noen forhåndsdefinerte overlays som kan benyttes for å illustrere prissoner. Dette er en KML fil med omrisset av en sone, som vi fargelegger etter prisen i den aktuelle sonen. I eksempelet under har vi tegnet opp overlays på Norden, men det er fullt mulig å lage overlays til alle prissoner. Dersom man har koordinatene for omrisset, og bruker riktig projeksjon på kartene, kan man importere overlays fra f. eks et SVG kart. Vi gjorde et forsøk med å importere forskjellige overlays fra Chart FX Maps, men dette fungerte ikke på grunn av Google Maps og kartgrunnlaget i Chart FX Maps benyttet seg av forskjellige kartprojeksjoner. Dette står beskrevet nærmere i Teknisk beskrivelse, se også figur 4.7 for illustrasjon.



Figur 4.4: Illustrasjon av overlay

Generere overlay Overlayene som ble vist i forrige avsnitt er tegnet med vår egen funksjon for å lage polygoner rett i Google Maps. Denne funksjonen ble laget med tanke for muligheten til å utvide med nye prissoner, denne funksjonen utelukker kartprojeksjonsproblemet.



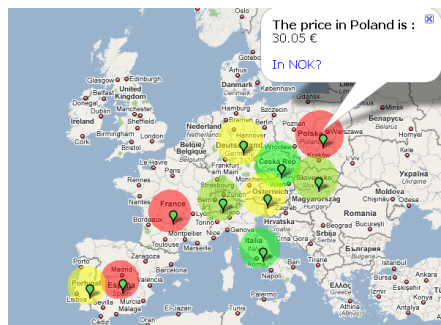
(a) Lage punkter

(b) Overlay i Google Earth

Figur 4.5: Manuelle overlays

Med denne funksjonen kan Navita lage alle de polygonene de måtte ønske, og bruke disse som en prissoner.

Generelle prissoner De manuelt tegnede overlayene som vi diskuterte i de to foregående avsnittene krever forarbeid for å kunne visualisere prissoner. For å kunne ha en mulighet for å illustrere prissoner uten noe forarbeid, og hvor som helst i verden har vi laget et løsningsforslag som ikke benytter seg av polygoner for å vise prissoner. Her benytter vi oss av sirkler som skal illustrere et prisområde. Ved å benytte oss av Geonames søker vi opp senterpunktet av f. eks et land, og tegner en sirkel rundt dette punktet. Fremgangsmåte for å legge til nye sirkler er meget enkelt. Man søker opp ønsket lokasjon, her henter vi svaret fra Geonames og tegner en sirkel.



Figur 4.6: Illustrasjon av generell overlay

4.2.2 Bruk av løsninger på tvers av teknologiene

Det er forskjell på de to teknologiene vi har jobbet med. På tross av dette har vi gjort forsøk på å lage begge løsningene så standardiserte som mulig. Vi har benyttet XML for lagring i begge løsningene, og vi har forsøkt å innføre samme koordinatsystem i begge løsningene. Begge løsningene har sterke og negative sider. Skrivebordsløsningen har et klart overtak på webløsningen når det gjelder markeringen av prissoner. Polygonet vi skal fargelegge er innebygd i kartet, mens på webløsningen må vi manuelt lage polygonet som vi skal markere som en prissoner.

Vi gjorde et forsøk for å forenkle arbeidet med å tilrettelegge nye prissoner for webløsningen. Teorien er å eksportere et polygon fra SVG kartet, og konvertere koordinatene til webløsningens GPS-koordinater. Lykkes vi i denne operasjonen vil vi ha en triviell jobb å lage polygoner (overlays) for nye områder. I neste avsnitt har vi forklart hvordan vi har gått frem for å konvertere polygoner fra et SVG kart til å benytte seg av GPS-koordinater. I teorien vil dette forenkle mye manuelt arbeid med å lage nye polygoner.

Konvertering fra skrivebordsløsning til webløsning

Dette avsnittet forklarer hvordan vi har hentet ut polygoner fra skrivebordsløsningen og benyttet dette i webløsningen. Vi har konvertert koordinatene i SVG filene til et geografisk koordinatsystem som baserer seg på lengde- og breddegrader, GPS-koordinater.

Dette ble gjort ved å velge ut to lokasjoner i SVG kartet hvor det lar seg gjøre å definere koordinatene. Vi brukte Lindesnes og Nordkapp, da disse stedene er lette å identifisere på et kart, kartgrunnet vi eksporterer fra har nemlig ikke geografiske referanser. Etter vi har bestemt oss for hvilke utgangspunkt vi ønsker

å benytte, finner vi stedenes GPS-koordinater. Ved hjelp av matematiske formler klarer vi å regne ut de resterende punktene for hele polygonet. Resultatet av denne utregningen er vist i figur 4.7.

I figuren kan vi se at resultatet ikke ble helt slik vi ønsket oss. Vi kan se et klart avvik i kartgrunnlagene. Selv om det kan se slik ut, behøver det ikke bety at et av kartene er feil. Grunnen til avviket er forskjell i kartenes projeksjon.

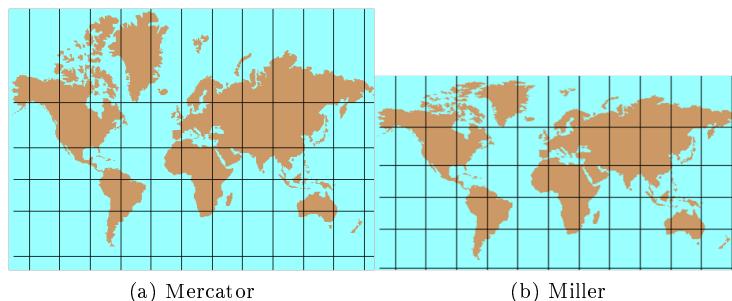


Figur 4.7: Prosjeksjonsfeil

Kartprojeksjoner Google Maps bruker ifølge Bret Taylor[2], Product Manager i Google, Mercator projeksjon. Mercator projeksjon er en form for sylindrisk projeksjon som gjør at linjene følger riktig kompasskurs. Projeksjonen som Chart FX sine SVG kart er basert på er uvisst. Andre Goncalves[1] i Chart FX support hadde ingen informasjon om projeksjonen som ble brukt på kartene. Chart FX har kjøpt kartene fra en tredjepart selskap, og de har fokus på fremstilling og ikke geografisk korrekthet.

Grunnen til at det er flere forskjellige projeksjoner er at det er vanskelig å lage et flatt kart av jordkloden, det er ingen flate projeksjoner som klarer å gjengi hele jordkloden geografisk korrekt. Dette er fordi jordklodens fasong er uregelmessig, men det er mulig å gjengi deler av den korrekt ved bruk av ulike projeksjoner. Som et eksempel på to typer projeksjoner kan vi se på Mercator og Miller projeksjon, som begge er sylindriske projeksjoner og hvor Mercator blir benyttet av Google Maps. Som vi kan se av figur 4.8 blir det mindre forvrenginger nær pol områdene ved bruk av Miller projeksjon. Dette er grunnen til at løsningen ikke endte opp med å overlappes med Google Maps sitt kartgrunnlag.

Det finnes ulike matematiske formler for å konvertere koordinatene fra en type projeksjon til en annen, men disse formlene er avanserte og tidkrevende å benytte seg av. Vi håpet på forhånd at løsningen ville konvertere alle mulige SVG filer til Google Maps sitt koordinatsystem, men siden det er mange forskjellige projeksjoner å forholde seg til ville dette krevd at løsningen måtte håndtere alle disse. Vårt prosjekt går ut på å se på mange ulike tekniske løsninger, og vi fant



Figur 4.8: Sylindriske projeksjoner

det ikke hensiktsmessig å gå dypere i problemet, da vi kan benytte oss av andre metoder for å lage overlays i webløsningen.

Konvertering fra webløsning til skrivebordsløsning

Vi har sett på muligheter for å bruke løsninger utviklet for skrivebordsløsningen som også kan brukes for webløsningen. I dette avsnittet skal vi se på mulighetene for å bruke løsninger utviklet for webløsningen som kan brukes i skrivebordsløsningen.

Informasjon som kan være interessant for oss er KML filene vi har laget i webløsningen (overlayene), og benytte disse som kartgrunnlag i skrivebordsløsningen. I teorien skal dette fungere da både KML- og SVG formatet er XML filer, og med små endringer kan vi oversette KML filene til SVG filer. Fordelen med dette er at vi får en skrivebordsløsning med geografisk korrekt kartgrunnlag. Baksiden vil være at KML filene som må lages for dette formålet må være svært detaljerte for at de skal fungere som kartgrunnlag, noe vi anser som en tidkrevende jobb. På en annen side er dette en engangsjobb som kun trengs å gjøres når nye områder skal legges til. Vi har dessverre ikke hatt mulighet til å forfølge denne muligheten, noe som vi mener er synd da dette kunne gitt oss et interessant resultat. En av grunnene til at vi ikke prioriterte å forfølge denne løsningen er problematikken rundt fleksibiliteten til Chart FX. Selv om vi bytter ut kartgrunnlaget vil vi fortsatt ha problemer med å legge til informasjon på kartgrunnlaget. En annen grunn til at dette forslaget ikke ble gjennomført er SVG formatets koordinat system.

Vi forsøkte å konvertere SVG's relative koordinater til GPS-koordinater. I teorien finnes det muligheter for å gjøre dette, men i praksis ble det verre. Etter gjennomført konversjon fra relative koordinater til absolutte koordinater, gjenstår det å konvertere de absolutte punktene fra SVG-koordinater til GPS koordinater. Konverteringen lot seg gjennomføre, men resultatet ble ikke slik vi håpet på. Da ett sett med GPS-koordinater kan være 5.009168, 57.3780391 taklet ikke editeringsverktøyet Adobe Illustrator disse koordinatene. Adobe Illustrator oppførte seg irrasjonelt og konverterte koordinatene ikke konsekvent og systematisk, men gjorde dette tilfeldig. Som resultat fikk vi punkter som var

direkte feil, og utseendet på polygonene ble ikke som ønskelig. Koordinatene ble automatisk konvertert fra GPS-koordinater til SVG koordinater. Resultatet av dette ble at konverteringsjobben vi gjorde var borte, og SVG koordinatene var ikke noe villig til å samarbeide. Vi mener at tankegangen bak konvertering til GPS-koordinater var smart, da hadde vi fått et geografisk korrekt kartgrunnlag, med gode muligheter for å legge til komponenter til korrekt geografisk lokasjon. Et annet problem med flate kart er kartprojeksjon som gjør at dersom vi bruker en type projeksjon i kartgrunnlaget, og legger til overlays med en annen type projeksjon vil man få feil resultat selv om koordinatene er de samme. Mer om dette står beskrevet i seksjon 4.2.2. (Se illustrasjon 4.7)

Et annet interessant forslag til å hente løsninger fra webløsningen til skrivebordsløsningen er selve kartgrunnlaget webløsningen benytter. Kartgrunnlaget til Google Maps er meget bra, og vi har vært inne på tanken å eksportere dette kartet til et format som vi kan benytte i skrivebordsløsningen, f. eks SVG. Grunnen til at dette ikke har blitt gjort er de samme som i forrige avsnitt, men her er det også en juridisk problemstilling rundt bruken av Google Maps. Under er det et utdrag fra Googles Terms of Service:

“5.3 Du samtykker til ikke å bruke (eller forsøke å bruke) Tjenestene til noen andre formål enn gjennom den grenseflaten som er levert av Google, med mindre du gjennom en separat avtale med Google har blitt gitt en særskilt tillatelse til å gjøre det.” [5]

Vi tolker dette punktet til at det ikke er tillatt å bruke Googles kartgrunnlag i noe annet format enn dets opprinnelige format, eller grenseflate. Dersom vi benytter Google Maps som vi har gjort i prosjektet mener vi at vi holder oss innenfor Googles regelverk. Hvis vi hadde tatt steget med å forsøke å eksportere kartgrunnlaget i Google Maps ville vi brutt Googles retningslinjer. Dette er ikke noe vi ønsker da vi gjør en oppgave for et seriøst foretak som planlegger å benytte vårt arbeid i fremtiden.

Kapittel 5

Brukerundersøkelse

For å kunne tilpasse applikasjonen til de ulike brukergruppene trengte vi et bedre innblikk i arbeidssituasjonen til en sluttbruker av de systemene som skal baseres på våre løsningsforslag. Det er flere måter å gjøre dette på, men på en eller annen måte må sluttbrukernes behov identifiseres. Dette er i seg selv et problem da det egentlig ikke finnes noen sluttbrukere av produktet enda, da produktet ikke er på markedet. Alternativet blir å undersøke hvem denne applikasjonen skal lages for. Når vi vet dette må vi finne ut hva disse brukerne forventer og krever av en slik applikasjon.

Hensikten er ikke at vi skal lage et ferdig produkt for sluttbrukere, vår leveranse skjer til Navitas programvareutviklere. Vi har likevel valgt å involvere sluttbrukerne, og vi får da et indirekte forhold til sluttbrukerne.

5.0.3 Halden Center of Expertise Energy Research

“Halden Center of Expertise Energy Research” har planer om å bli et verdensledene utdanningsmiljø for energiforskning med studenter fra hele verden. Institutt for Energiteknikk, Høgskolen i Østfold, Universitetet for miljø- og biovitenskap og bedrifter som Navita, Nasdaq OMX, Tieto Enator, Saab og Nexans er alle involvert i prosjektet. Senteret vil bli plassert ved siden av dagens Høgskole bygg i Halden. Senteret har som mål å hente 5 professorer, 10 post-doc forskere, 10-15 doktorgradstipendiater og 250 studenter på masternivå. Senteret skal utvikle et analyse- og prognosesenter for markedsbasert salg av kraft, gass, olje, klimakvoter etc. Høgskolen i Østfold skal lede arbeidet med visualiseringsteknologi, system engineering og informasjon og kommunikasjon ved det nye senteret. [35] [6] [24]

Selv om senteret ikke er operativt med eget bygg enda har de satt i gang prosjekter. Våre intervjuobjekter består av to personer som har sine oppgaver tilknyttet denne satsingen. Medlemmer i gruppen er en doktorgrad stipendiat fra Høgskolen i Østfold, og en masterstudent ved Universitetet for miljø-og biovitenskap i Ås. Disse personene jobber i likhet med oss i samarbeid med Navita som er meget sentrale i satsingen for det nye utdanningstilbudet i Halden. Gjen-

nom deres arbeid har de fått kjennskap til fremtidige sluttbrukere, og deres daglige arbeidsoppgaver. Gruppen hjalp oss med å identifisere behov og fastsette krav.

5.0.4 Metode

Vi refererer til interaksjonsdesigns fremgangsmåter for å komme fram til et design som gagnar brukeren mest. Design er mer enn fargevalg, og plassering av objekter. Helheten i systemet planlegges i designfasen. Balanseringen mellom funksjonalitet og brukervennlighet er en viktig del av designfasen. Det er fire grunnleggende steg som må gjøres for at et endelig design skal bli så godt som mulig.

Datainnsamling

Det finnes ulike måter å samle informasjon om brukerne. Et vanlig scenario er at man har definert en målgruppe for hva man skal utvikle. Når man har definert en målgruppe er neste steg å innhente informasjon fra målgruppen. I denne operasjonen bør man finne den mest egnede måten å innhente data. Skal man utvikle noe med en ganske stor målgruppe, f. eks ungdom mellom 16 og 20 år kan det lønne seg å lage en spørreundersøkelse med avkryssingsspørsmål. I vårt tilfelle har vi en mindre potensiell brukergruppe. Det er lite hensiktsmessig å lage et spørreskjema for utsending i stor skala. Fordi det er en ganske omfattende og forklare et potensielt system og fordi det neppe er mulig å identifisere et tilstrekkelig stort antall brukere. Det ville ta lang tid å få data tilbake i forhold til den tiden vi har til rådighet.

Et alternativ til spørreundersøkelse er intervju. Et intervju tar lengre tid dersom man skulle intervjuer mange mennesker. Da vår gruppe er liten, og det rekkes med en intervjurunde er dette et godt alternativ, og i realiteten det eneste reelle alternativet. Fordelen med et intervju er at man kan stille spørsmål hvor intervjuobjektet kan svare bredere/dypere enn de forhåndsbestemte svaralternativene til et spørreskjema. Vi kan stille mer spesifikke spørsmål tilpasset akkurat den brukeren vi intervjuer. I et intervju har også intervjuobjektet mulighet til å komme med innspill som intervjuerne kanskje ikke hadde tenkt på.

Det finnes forskjellige måter å gjennomføre intervju på. Fra boken "Interaction Design", [27] finnes det tre forskjellige måter:

- Ustrukturert: hvor man ikke er ledet av et manus. Man har muligheter for å stille de spørsmål man vil der og da, men er lite egnet for å repetere intervjuer til andre brukere.
- Strukturert: Følger nøye et manus, nesten som et spørreskjema. Her kan man risikere å ikke få etterfulgt interessante kommentarer fra intervjuobjektet da man følger et manus. Lett å reproducere i flere intervjuer.
- Semistrukturert: Intervjuet er basert på et manus, men man har mulighet for å forfølge interessante kommentarer som kommer fram under sesjonen. Kan være en god kombinasjon mellom dybde og reproduksjonsvennlighet.

Vi har basert oss på å bruke et semistrukturert intervju. Vi har laget et manus med spørsmål som er interessante for oss, men lar intervjuobjekter komme med innspill og kommentarer. Med denne formen for intervju har vi mulighet for å forfølge interessante utspill fra intervjuobjektene. Dette er en fordel da grunnen til at vi gjennomfører intervjuet er for å høre brukernes meninger og forslag. Rent metodisk har dette også sine fordeler at brukerne føler seg hørt i utviklingsprosessen kan være en fordel. Brukerne føler et eierskap til løsningene og det kan være større sannsynlighet at de vil ta de nye løsningene godt i mot. [27]

Våre arbeidsoppgaver

- Identifisere behov, fastsette krav
- Utvikle flere typer design
- Lage interaktive prototyper
- Evaluere design

Selv om vi ikke skal lage et ferdig produkt, er det nyttig å involvere brukerne så tidlig som mulig. Når våre forslag til løsninger er implementert med tanke på brukerne, og brukerne har blitt involvert på et tidlig tidspunkt vil det være større sjanse for at bruker hensynene blir ivaretatt, enn om vi ikke hadde gjort en innsats for å implementere brukernes ønsker. Hvem er brukeren og interessenter av systemet? Dette er ikke så enkelt som det ville vært med en applikasjon med et klart funksjonelt mål. Brukeren av systemer som bygger på våre løsninger kan være:

- De som direkte jobber med løsningen.
- De som administrerer brukerne av løsningen.
- De som gjør beslutninger på bakgrunn av løsningen.

De overnevnte brukerne kan sammenlignes med Easons (1987) inndeling av et systems brukere:

- **Primærbrukere:** De som direkte bruker produktet på jevnlig basis.
- **Sekundærbrukere:** De som bruker produktet av og til, eller får kjennskap til innholdet av andre.
- **Tertiærbrukere:** De som blir berørt av produktets introduksjon, eller vil påvirke ens handlinger.

Fra dette kan vi si at flere enn de som daglig bruker produktet blir påvirket av produktets introduksjon i omgivelsene. Vår oppgave er først å fremst å konsentrere oss om primærbrukere, og til dels sekundærbrukere. Sekundærbrukere kan

være aktive brukere av systemet, men til mindre grad enn primærbrukere. Med andre ord så kan vi si at direktebrukerne av produktet er i vår hovedmålgruppe. Men man må også være klar over at direktebrukernes hverdag kan påvirke ens medarbeidere og ledere.

Basert på den teorien vi nå har beskrevet må vi finne ut av hvem brukerne er, brukernes krav, lage en modell eller prototyp som tilfredsstill brukernes krav.[28] [27]

Dette blir en slags annerledes brukerundersøkelse og er slikt sett ikke ideell. Fordelen er at vi samarbeider med det som nå er vår primærmålgruppe: programvareutviklere.

Identifisere brukere, og brukernes krav

Vår kilde er medarbeidere i prosjektet ved Halden “Halden Center of Expertise Energy Research” som står omtalt i seksjon 5.0.3. Vår kilde har kjennskap til eventuelle brukere av systemet og kan hjelpe oss med å identifisere brukere og fastsette krav.

Spørsmål som ble stilt til prosjektgruppen:

- Hvem er brukeren?
- Hva vil systemet brukes til?
- Hva er brukernes krav/ønsker til systemet?
- Hva er viktigst for brukeren?
- Hvilke beslutninger skal tas på bakgrunn av systemet?

Av disse spørsmålene fant vi at følgende brukere er aktuelle for dette systemet: Kraftmegler, nettoperatør(logistikk), analytiker og forsker. Hva som er viktig for de forskjellige brukergruppene står beskrevet i problemstillingen, se seksjon 2.1.2.

Vi ønsket å finne ut hva som er viktig for de ulike brukergruppene i deres daglige arbeidssituasjon og hvilke beslutninger som blir tatt basert på informasjon som blir hentet ut av applikasjonen.

Tverrfaglighet

Vi oppdaget under intervjuet at brukere av denne applikasjonen kommer fra flere fagfelt. Det er ingen utpregete “gass-folk”. Produsentene har gass som fagfelt, nettoperatørene kan gass og transport/logistikk. Resterende brukergrupper som kraftmeglere, forskere og analytikere trenger ikke ha sitt fagfelt i gass, da disse gruppene bruker dataene fra applikasjonen som grunnlag for sine beslutninger.

Kraftmegler

En kraftmegler er avhengig av å kunne ta raske beslutninger. Kraftmeglere er kortsiktige og er mest opptatt av priser, prisendringer og nyheter. Kraftmeglere får sin inntekt fra kjøp og salg av kraft. Derfor er kraftmeglernes avhengig av hyppig oppdatert informasjon og muligheter til å handle raskt. Det vil si minst mulig interaksjon, mest mulig relatert informasjon, men samtidig lettlest grafikk. Målet for en kraftmegler er å kjøpe billig og selge dyrt. Pr i dag bruker kraftmeglere tabeller, og er egentlig fornøyd med det. Fordelen med grafikk er å kunne få et oversiktsbilde som kan være vanskelig å oppdage i tabellform spesielt for hendelser som meglernes ikke leter spesifikt etter i sine daglige rutiner.

For en kraftmegler er en rørledning som blinker en god nok indikator på at det har skjedd noe med rørledning/knutepunkt. Det kan ta lang tid fra et brudd skjer, til konsekvensen rammer kundene. Dette gjelder spesielt på etapper med lang overføringstid.

På de lange transportetappene, som f. eks. rørledninger fra andre kontinenter kan man først merke konsekvensene flere timer etter at en ulykke har skjedd. Med et grafisk verktøy der man har oversiktsbilder over deler av kraftnettet man vanligvis ikke jobber i, har man en større mulighet for å forhindre at konsekvensene etter en ulykke har innvirkning på sluttbrukerne av kraften. Dette kan f. eks. gjøres ved å hente kraft fra andre områder enn vanlig.

For å oppdage endringer som kan føre til at en handling fra kraftmeglernes side er blinking i oversiktiskartet en god nok alarmindikator. Hvis man tenker seg standard konfigurasjon, med maks kart synlig i bildet. (zoomet helt ut) er det komplisert å få plassert inn detaljert informasjon på grunn av begrenset plass, (beskrevet i detaljeringsgrad seksjon 3.4.6). Her vil bare de største rørledningene være synlig. Men dersom noe skjer, kan det blinke i kartet slik at brukeren blir oppmerksom på at det har skjedd noe. Da må brukeren gjøre noe, enten ved å klikke for å etterforske nærmere for å endre detaljeringsgrad, slik at brukeren bestemmer selv om denne hendelsen er interessant for vedkommende eller ikke.

Et grafisk verktøy for en kraftmegler vil være veldig bra, da en kraftmegler ikke klarer å holde oversikten over større områder, eller områder utenfor deres kjente områdene ved å lese tabeller. En kraftmegler har sine faste ting som de ser etter i tabellene, og en sjelden hendelse vil kanskje ikke komme på de faste tabellene før det er for sent, og dermed så mister kraftmeglernes muligheten for å gjøre profitt ved å være føre var og tilby brukerne kraft fra alternative kilder.

Forsker, analytiker

Forskere og analytikere har ganske like roller i vår sammenheng, derfor er de slått sammen som en bruker med de samme kravene. I de tilfeller der de har forskjellige roller vil vi beskrive dette nærmere.

Denne brukergruppen er ute etter en så detaljert informasjon som mulig. All data er interessant for en forsker og/eller analytiker. Priser, historikk, hendelser, trender og mengder med data. Denne brukergruppen vil ha informasjon både grafisk og i tabellform. Denne brukergruppen har god tid, og er grundig i sitt

arbeid. Lettlest grafer ikke så viktig for denne brukergruppen. Presisjon er det viktigste. Hendelseslogg, problemområder og trender er viktige stikkord. Forskjellene mellom en analytiker og forsker i denne sammenhengen er tidshorisonten de jobber i. En analytiker har som oppgave å “spå” mulige scenarioer, mens en forsker gjerne har en tidshorisont på flere år og viser til forandringer i markedet og dens konsekvenser.

Beslutningene som tas av denne brukergruppen er på et mer rådgivende plan. Denne brukergruppen selger eller kjøper ikke kraft, men gir råd til sine arbeidsgivere, enten det er et kraftmeglingselskap, gassprodusenter, nettoperatører, politikere osv. Forskere vil utrede eventuelle problemer med gassflyten, for eksempel om det er forskjell i gasstransporten i de ulike årstidene. Forskere bruker datagrunnlaget fra applikasjonen på lang sikt for å utrede og komme med forslag til løsninger på eventuelle problemer.

Denne brukergruppen vil være primærbrukere eller sekundærbrukere av systemet. Forskeren eller analytikeren kan hente ut data de selv bruker, dette gjør de til primærbrukere dersom de benytter seg av systemet på jevnlig basis. Dersom forskeren eller analytikeren bruker applikasjonen av og til, eller velger å bruke informasjon som er hentet ut fra systemet fra en annen bruker, kan de kalles sekundærbrukere.

Nettoperatør (gass-logistikk)

Nettoperatørene og produsentene av gass har et nært samarbeid. Produsentene er avhengig at nettoperatøren får fraktet gassen til etterspørrende områder. En nettoperatør vil ha stor nytte av en grafisk fremstilling av gassnettverket. Her vil en operatør typisk velge å vise fram sin del av gassnettverket, og raskt finne ut dersom det er problemer med overføringen av gass, eller eventuelle underleverandører med problemer.

Nettoperatørens viktigste interessefelter er flaskehals, kostnader og inntekter. Her er det viktig å knytte visualisering til kostnader og inntekter under transport av gass. Det må være lett å hente ekstra informasjon, som:

- Trykk
- Pris
- Flyt

Denne informasjonen kan fremvises etter initiativ fra brukeren. Det er ikke hensiktsmessig å vise detaljert informasjon, dersom verdiene er i intervallet for normal drift. Det er først ved unormale verdier at brukeren har behov for dette. En mulig løsning for å tilfredsstille brukernes ønske er å gjøre brukeren oppmerksom på at det har skjedd noe, men la brukeren selv ta initiativ for å få fram informasjonen.

Nettoperatører kan tilhøre alle gruppene til Easons inndeling av brukere, primær- sekundær- og tertiærbrukere. De personer som har oppgaven med å

overvåke kraftnettverket og som bruker applikasjonen jevnlig, vil være primærbrukere. De som bruker applikasjonen av og til, eller får kjennskap til informasjon fra en primærbruker vil være sekundærbrukere av applikasjonen. Andre deler av bedriften, f. eks ledelsen kan være tertiærbrukere av applikasjonen da de ikke bruker den selv, men baserer avgjørelser på bakgrunn av informasjon hentet ut fra applikasjonen.

Konklusjon fra brukerundersøkelsen

Undersøkelsen med denne prosjektgruppen som hadde kunnskap om eventuelle sluttbrukere til applikasjonen var svært nyttig for vår del. Før dette møtet var vi svært usikre på bruksområder og brukere til applikasjonen utover kraftmegling.

Alt peker i retning av at applikasjonene som skal utvikles kan med fordel baseres på samme plattform. Kravene til de forskjellige brukerne bør kunne aktiveres/deaktiveres etter behov.

Kapittel 6

Tekniske løsninger

Dette kapittelet inneholder de løsningene som har blitt overlevert til Navita. Prosjektet har basert seg på å utvikle prototyper som vi har diskutert i samarbeid med Navita. Diskusjonene har medført endringer i kravene og hvilke løsninger som skal prioriteres, og dette har medført at noen prototyper har blitt kassert. De løsningene som er beskrevet videre i dette kapittelet er ikke å regne som komplette løsninger, men heller prototyper for å vise potensial i de ulike verktøyene.

Datagrunnlaget er basert på data som er hentet fra Nord Pool Spot og fiktiv testdata. Alle løsningene benytter seg av XML-standarden, og på bakgrunn av at Navita ikke overleverte XML-skjemaet for deres data, laget vi vårt eget XML-skjema (seksjon 2.1). Vi kan derfor ikke være sikre på at datagrunnlaget vi har lagt til rette for løsningene kan gjenspeiles i Navitas datagrunnlag, men de bør være omtrentlig like og det vil ikke kreve store inngrep for å tilpasse løsningene for Navitas data.

6.1 Skrivebordsløsning

Skrivebordsløsningen er utviklet i Visual Studio.net 2005 med .net Framework 2.0 og ved bruk av Chart FX Maps. Kravene til løsningen er forklart i oppgavebeskrivelsen og omfatter:

1. Visualisere prissoner på kartgrunnlag.
2. Visualisere tidsperspektiv på forskjellige prisområder.
3. Ha mulighet for å zoome til prisområder.
4. Visualisere rørledninger, noder med logistikk info.
5. Vise flyt mellom prissoner.
6. Visualisere smart-mapping, metrokart.

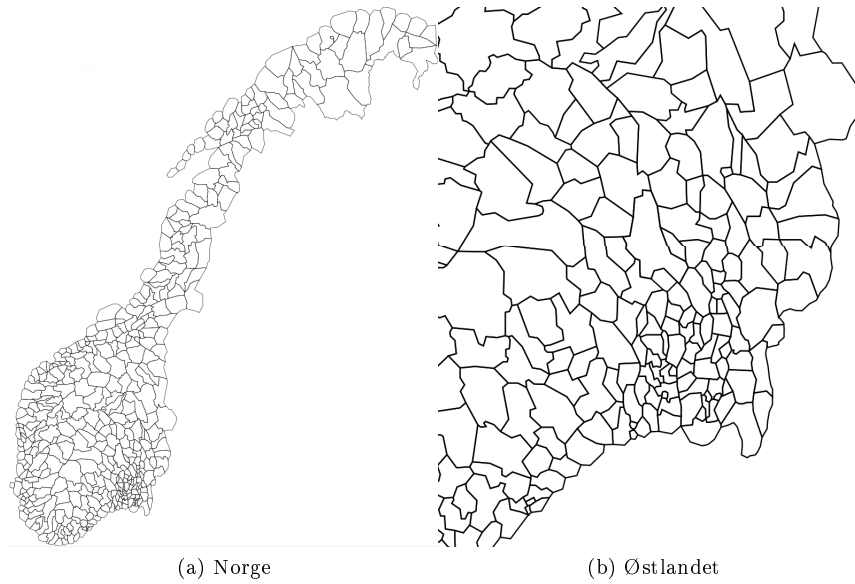
Skrivebordsløsning har funksjonalitet for å visualisere prissoner, tidsperspektiv og zoom. Dette vil si at datagrunnlaget vi har hentet fra Nord Pool Spot, som inneholder prisinformasjon for de nordiske områdene over et døgn, skal representeres på kartgrunnlaget. Representasjonen er i form av fargekoding av områdene, i henhold til prisnivå, og bruk av tekst som viser nøyaktig pris. Tidsperspektivet er implementert ved bruk av en forløpsindikator som gir mulighet for å bla seg gjennom timene. Zoom funksjonen trenger ikke å bli utviklet da den allerede er innebygd i Chart FX, den gir mulighet for å zoome inn på et polygon. (prissone)

Løsningen som er utviklet benytter seg av to eksterne filer:

SVG Kartgrunnlaget baserer seg på SVG standarden.

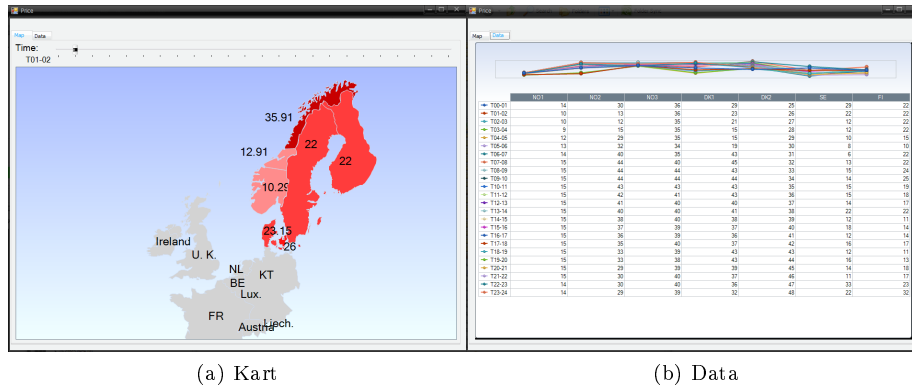
XML Datagrunnlaget inneholder prisinformasjonen for områdene i XML format.

Kartgrunnlaget som Chart FX benytter seg av er basert på SVG standarden, som igjen må følge Chart FX sin egen standard for oppsett av SVG filene. Det vil si at man ikke står fritt til å benytte seg av egne SVG filer, da Chart FX ikke vil klare å håndtere filene korrekt. Hovedsaklig er det grupperinger av polygoner som Chart FX er avhengig av, da det er disse som gir mulighet for å koble statiske data opp mot kartgrunnlaget. Problematikken rundt dynamisk soneinndeling ble forsøkt løst ved bruk av et kommunekart av Norge, slik at man hadde mulighet for å dele inn Norge i tre dynamiske prissoner ved bruk av kommunepolygonene. Men da dette kartgrunnlaget ikke på langt nær fulgte Chart FX standarden fant vi det ikke hensiktsmessig å konvertere dette, noe Navita var enig i. I tillegg mente Navita at kommunekartet ikke representerte Norge visuelt godt nok. Dette kan vi se på figur 6.1 hvor man ser at fjordene ikke blir representert da kartet viser de kommunale grensene og ikke fastlandet.



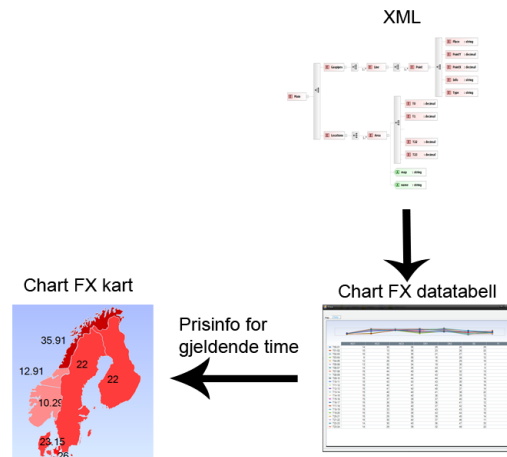
Figur 6.1: Kommuneinndelt SVG

Datagrunnlaget er, som nevnt tidligere, basert på data fra Nord Pool Spot. Dataene inneholder prisinformasjon for hvert område i Norden over et døgn, inndelt i timer. Denne informasjonen ble lagret som en XML fil da dette gjør det enkelt å koble dataene opp mot Chart FX sine tabeller.



Figur 6.2: Skrivebordsløsning hovedbilder

Løsningen som ble utviklet er vist i figur 6.2a. Når løsningen starter opp vil den laste inn dataene fra XML filen inn i Chart FX sin datatabell, som man



Figur 6.3: Skisse av skrivebordsløsningen

kan se på figur 6.3. Brukeren har mulighet for å direkte editere denne tabellen, men det vil ikke påvirke XML datagrunnlaget. Kartet leser data for en og en time fra datatabellen og presenterer disse i kartet. Når brukeren endrer tidsperspektivet ved hjelp av forløpsindikatoren ovenfor kartet vil løsningen oppdatere kartet med nye data fra datatabellen. Dette vil automatisk oppdatere prisinformasjonen som er vist i kartet, som fargekoding og prisnivå teksten. Brukeren har i tillegg mulighet til å benytte seg av den integrerte zoom-funksjonen som finnes i menyen ved å høyreklikke på et polygon. Dette vil forstørre polygonet slik som vist på figur 6.4. Denne løsningen tilfredsstiller tre av de seks kravene



Figur 6.4: Demonstrasjon av zoom funksjon i skrivebordsløsningen

som var stilt, de resterende kravene har ikke blitt implementert. Hovedgrunnen til dette er at Chart FX ikke støtter presentasjon av dynamisk kartdata, noe som står beskrevet nærmere i konklusjonen, seksjon 7.1.4.

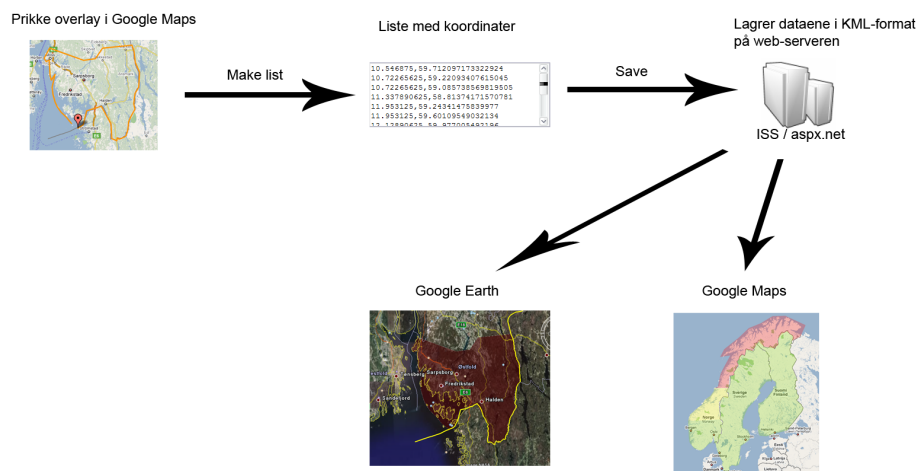
6.2 Webløsning

Webløsningen er hovedsaklig utviklet ved bruk av JavaScript og Google Maps API, men i tillegg er ASPX.NET, JSON, HTML og XML benyttet. Kravene til løsningen er tilsvarende til skrivebordsløsningen og de omfatter:

1. Visualisere prissoner på kartgrunnlag.
2. Visualisere tidsperspektiv på forskjellige prisområder.
3. Ha mulighet for å zoome til prisområder.
4. Visualisere rørledninger, noder med logistikk info.
5. Vise flyt mellom prissoner.
6. Visualisere smart-mapping, metrokart.

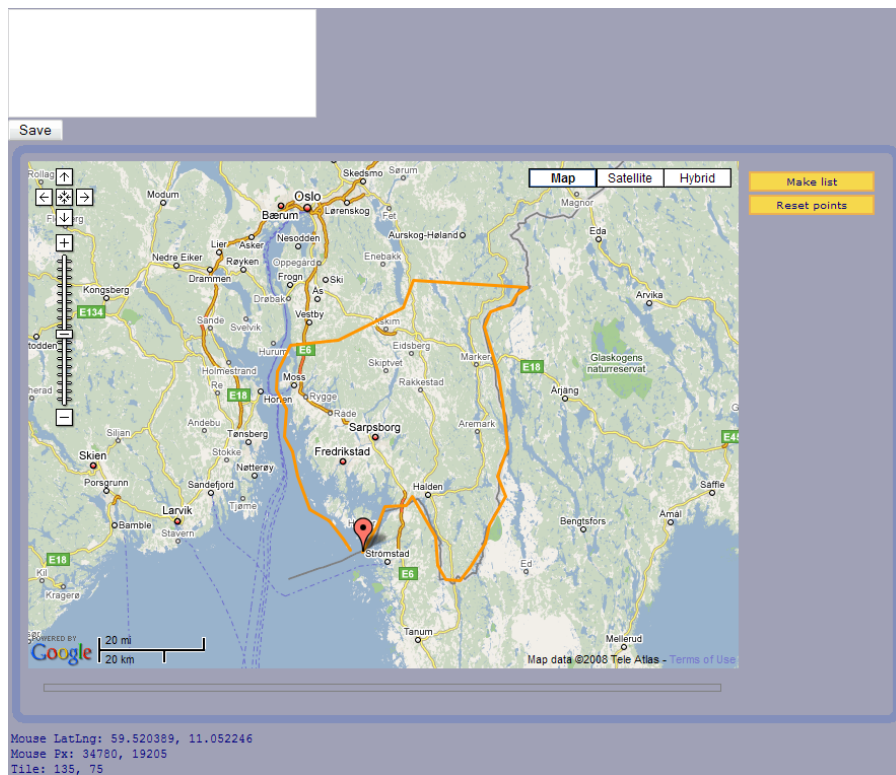
6.2.1 Inndeling av prissoner

Under prosjektets gang har vi lett etter løsninger for å dele inn prissonene dynamisk i Google Maps, som å benytte grenseinformasjonen som ligger i Google Maps og konvertering av eksisterende SVG kartgrunnlag. Grenseinformasjonen fant vi ut var utilgjengelig og konverteringen var ikke vellykket pga ulike kartprojeksjoner. Vi fant det derfor hensiktsmessig å opprette prissonene direkte i Google Maps, hvor man får inndelingen nøyaktig slik man ønsker. Dette er en manuell prosess som må utføres for hvert område man skal presentere, men for de fleste områder vil dette være en engangsprosess. Her har vi laget et verktøy for å lage prissoner. Den utviklede løsningen er vist i figur 6.6 hvor man ved hjelp av museklikk oppretter en linje som skal representere prisområdet. Etter endt optegning av polygonet velger man “Make list” som eksporterer koordinatene for linjen til tekstboksen. Dette gir mulighet for å benytte seg direkte av koordinatene, men det er også mulighet til å lagre koordinatene i en KML-fil på webserveren ved bruk av “Save” knappen, som er vist i figur 6.5.



Figur 6.5: Skisse av hvordan man oppretter prissoner

Når man har laget polygonene kan disse bli brukt for å representere prisnivået på tilsvarende måte som skrivebordsløsningen.

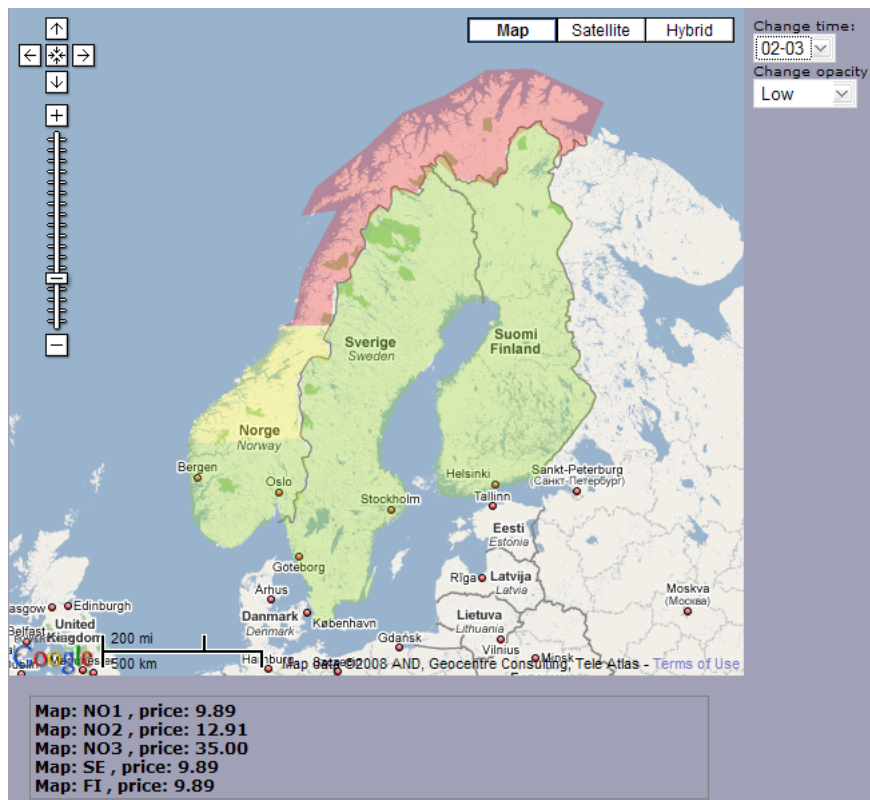


Figur 6.6: Verktøy for inndeling av prissoner

6.2.2 Visualisere prissoner

Visualiseringen av prissonene blir vist ved bruk av både fargekoding og tekst, en rullegardinliste tar seg av tidsperspektivet og muligheten for zoom er innebygd i Google Maps.

Løsningen, som er vist i figur 6.7, gjengir prisenivået for Norge, Sverige og Finland på en gitt time. Brukerinteraksjonen gir mulighet for å endre tiden og opasiteten på polygonene ved bruk av to rullegardinlister. Prisinformasjonen blir hentet fra en ekstern XML fil som inneholder data for alle polygonene over et døgn, når brukeren endrer tiden vil løsningen fjerne de eksisterende polygonene og tegne de opp på nytt siden fargekodingen kan ha blitt endret. Den nøyaktige prisen på hvert område blir vist som tekst på undersiden av kartet. I tillegg til funksjonene som er implementert tilbyr Google Maps noen innebygde funksjoner, deriblant zoom, panorering og endring av kartmodus.

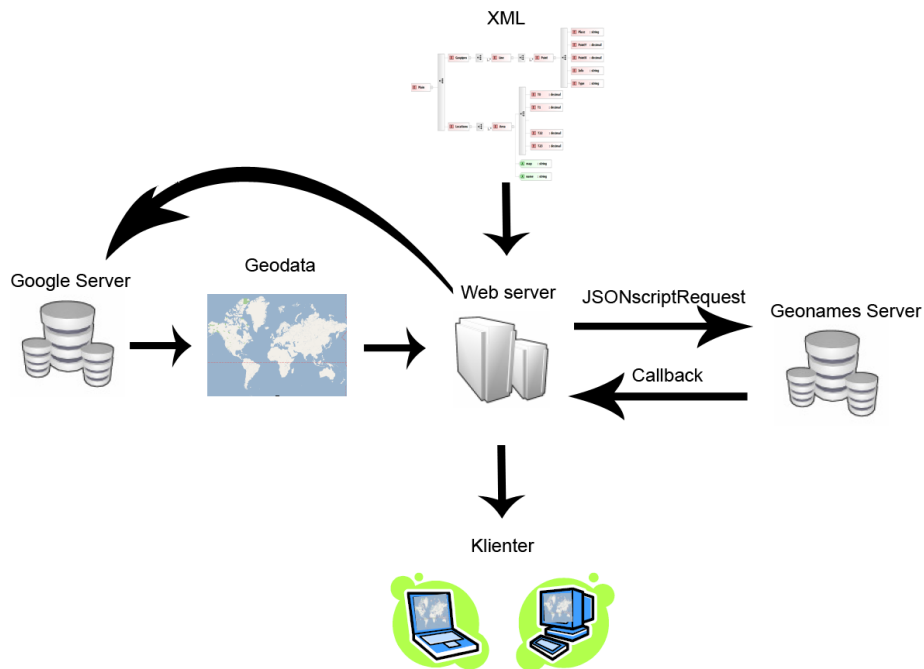


Figur 6.7: Visualisere prissoner

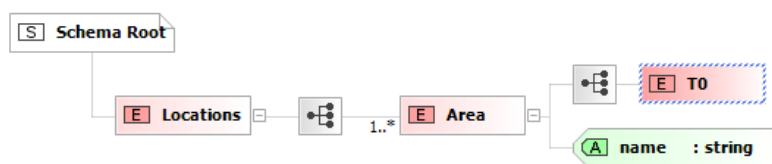
6.2.3 Generelle prissoner

Den neste løsningen vi har benyttet oss av går ut på å dele inn områder ved bruk av sirkler, for deretter å angi en pris til hver sirkel. Dette gir en grovere inndeling enn det som ble laget i visualisering av prissoner, men fordelene med denne løsningen er at den er rask å implementere og kan benyttes verden over. Den vil nok ikke kunne benyttes av alle målgruppene av systemet, til det blir den for grov, men de som trenger et oversiktsbilde av prisene vil denne løsningen muligens være tilstrekkelig. Løsningen baserer seg på input fra en XML fil som er beskrevet i figur 6.9, hvor hvert område blir beskrevet av et navn og pris per time. Prisene blir lagret i XML elementene T0, T1, T2 osv, figuren viser kun T0 på grunn av plassbesparing. Vi laget en skisse, figur 6.8, som viser hvordan løsningen henter ulike data. Webserveren henter data fra XML filen, hvor lokasjonsnavnet blir sendt til Geonames via et XMLHttpRequest for å motta GPS koordinater for lokasjonen. I mellomtiden blir det sendt en forespørsel til Google serveren for å hente riktig kartutsnitt. Forespørselen fra Geonames er asynkron og benytter en callback funksjon for å behandle dataene på webser-

veren. Når GPS koordinatene blir mottatt henter løsningen prisinformasjon fra XML filen for gjeldene lokasjon og oppretter en sirkel rundt koordinatpunktet og fargelegger denne i henhold til prisnivået, i tillegg blir den eksakte prisen koblet til senterpunktet av sirkelen ved å benytte informasjonsbokser som vist i figur 6.11. Den endelige løsningen vises i figur 6.10.

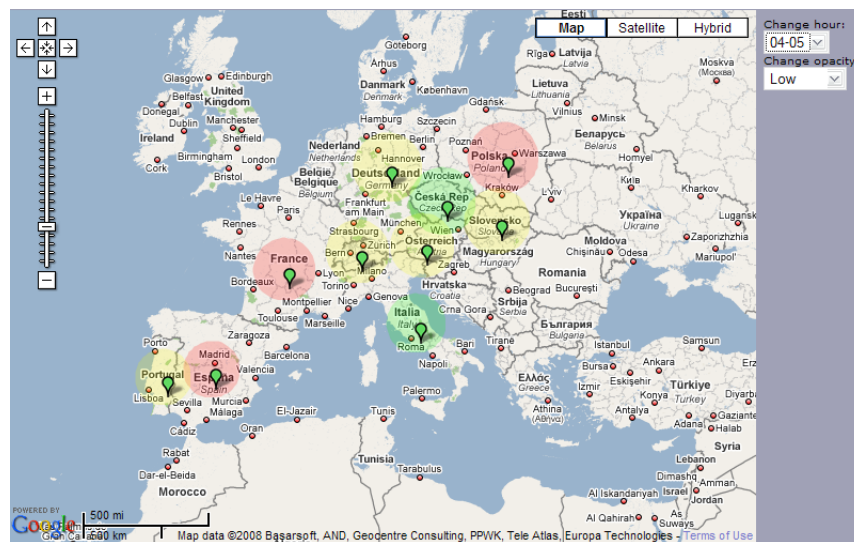


Figur 6.8: Skisse av generelle prissoner



Figur 6.9: XSD av Generelle prissoner

Brukerinteraksjonen gir mulighet for å endre tidsperspektivet og opasiteten ved bruk av rullegardinlister. Når brukeren endrer tiden vil løsningen fjerne de eksisterende sirklene og hente ny informasjon fra XML filen.



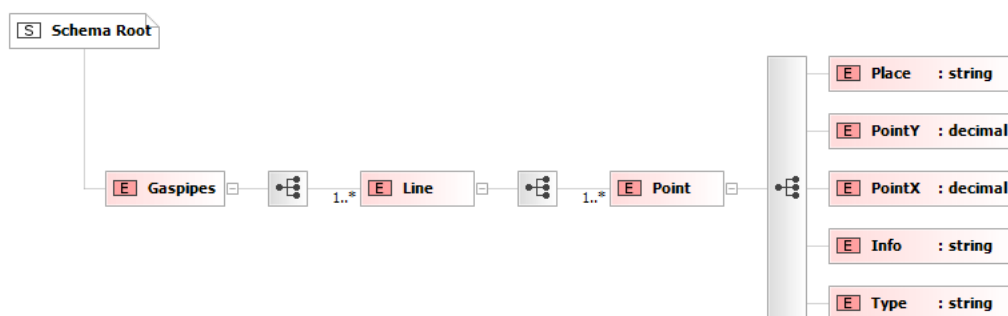
Figur 6.10: Demonstrasjon av generelle prissoner



Figur 6.11: Demonstrasjon av generelle prissoner prisinformasjon

6.2.4 Visualisere rørledninger

Visualisering av rørledninger innebærer å vise logistikken av kraft, dette være seg gass, olje, elektrisitet etc. For å vise logistikken skal vi benytte ulike elementer som f. eks. oljeplattformer, raffineringsanlegg, generatorer etc., og de fleste av disse elementene skal knyttes sammen med ledninger. Til de ulike elementene og rørledningene skal det knyttes ulik informasjon, fra eierforhold til kraftnivå og flyt. I Chart FX fant vi ingen god løsning på dette, men i Google Maps vil dette være mulig da Google Maps har støtte for å presentere dynamiske data på kartgrunnlaget. Løsningen benytter seg av en ekstern XML fil som inneholder informasjon om hvert element, slik man kan se av XSD filen på figur 6.12.



Figur 6.12: XSD som brukes for visualisering av rørledninger

Gaspipes Dette er grunnelementet som kan inneholde en eller flere Line elementer.

Line Line elementet er til for å dele opp de ulike ledningene mellom punktene. Line kan inneholde en eller flere Point elementer og alle Point elementene som ligger innenfor et Line element blir koblet sammen med ledninger.

Point Point inneholder informasjon om hvert punkt, det være seg oljeplattformer, raffineringsanlegg etc.

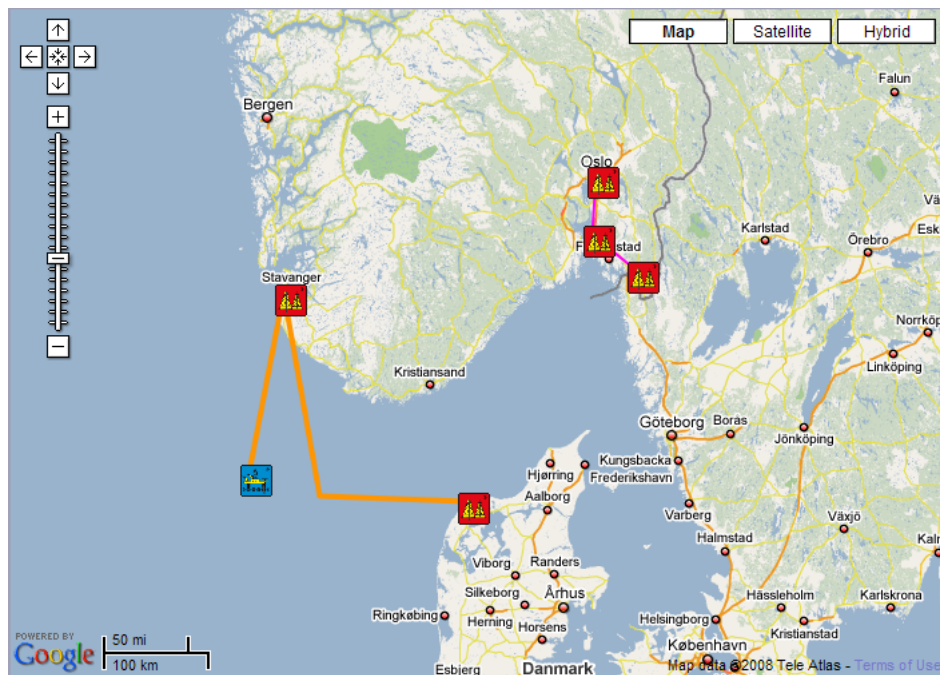
Place Navnet på lokasjonen.

PointY Breddegraden for lokasjonen i desimaltall.

PointX Lengdegraden for lokasjonen i desimaltall.

Info Informasjon tilknyttet punktet.

Type I oppgaven har vi valgt å benytte oss av to typer elementer, olje installasjon på land eller vann. Type-attributtet beskriver hva slags element det er og dette blir brukt for å koble riktig ikon opp mot elementet i Google Maps. Det er også en tredje variant av Type for å beskrive en knekk på ledningen, dette er et vanlig punkt som inneholder kun bredde- og lengdegrad informasjon.



Figur 6.13: Visualisere rørledninger demonstrasjon

Løsningen laster inn XML-filen og viser elementene slik det er vist i figur 6.13. Brukeren har deretter mulighet for å hente ytterligere informasjon om hvert punkt ved å holde musepekeren over ikonene, man vil da få opp en informasjonboks som er vist i figur 6.14 . Ikonene som blir presentert er i tillegg koblet



Figur 6.14: Informasjon knyttet til elementene

opp mot zoom nivået på Google Maps, slik at ikonene blir mindre desto lavere zoom nivået er, dette for å forhindre at ikonene dekker store områder ved lavt zoom nivå. Detaljeringsgraden vil endre seg ved ulike zoom nivåer. I starten av utviklingsprosessen for visualisering av rørledninger hadde vi implementert mulighet for å flytte ikonene med tilhørende ledninger, dette var ment som en

løsning til å øke lesbarheten for overlappende elementer, men Navita ønsket heller å prioritere utvikling av et geografisk korrekt system med mulighet for knekk på ledningene.

De ulike webløsningene tilfredsstiller fire av seks kravene som var stilt, smart-mapping og visning av flyt har ikke blitt implementert. Navita ønsket å prioritere visning av geografisk korrekte elementer og så på smart-mapping som en mindre nødvendig del av systemet. Visningen av flyt fant vi ingen tilsvarende mulighet for å implementere slik det er gjort i Nord Pool Spot. Det er mulig å vise flyt i informasjonsboksene som er koblet opp mot hvert element, da disse benytter standard HTML kode, som gir mulighet for å presentere grafer, bilder etc.

I tillegg til den tekniske beskrivelsen i dette kapitlet inneholder oppgaven et vedlegg (Tillegg A) som beskriver hvordan hver løsning fungerer og hvilke filer som blir anvendt. Dette tekniske dokumentet er laget for Navita, som et hjelpemiddel til både å eksekvere og videreutvikle de ulike løsningene. Navita sin utviklingsavdeling er flerspråklig og det var derfor et krav om at denne dokumentasjonen var på engelsk.

Kapittel 7

Konklusjon

I dette kapitlet vil vi gå gjennom løsningene vi har kommet fram til, hva vi har gjort og hva vi ikke har gjort. Som en del av konklusjonen vil vi gi våre vurderinger av løsningene.

Datastruktur

Forslaget til løsning som har blitt utviklet har tatt høyde for Navitas ønsker om datastruktur. Løsningen skal kunne knyttes til eksisterende eller planlagt infrastruktur slik at oppdatering av data og informasjonsflyt vil skje med så lite manuelle operasjoner som mulig. Med en fullstendig XML-skjema beskrivelse som grunnlag fra begynnelsen av prosjektet ville tilkoblingen til infrastrukturen være en triviell arbeidsoppgave. Det ble utviklet en midlertidig skjema-beskrivelse som vi mener er fornuftig for denne løsningen. Denne må i ettertid tilpasses for å passe i Navitas infrastruktur.

7.1 Skrivebordsløsning

Forslaget til skrivebordsløsning ble laget i Visual Studio 2005 og Chart FX Maps 7.0. Denne oppgaven baserer seg på å lage en versjon av Nord Pool Spot i disse verktøyene. Nøkkelfunksjoner (kravspesifikasjon) i denne løsningen vil være:

1. Visualisere prissoner på kartgrunnlag.
2. Visualisere tidsperspektiv på forskjellige prisområder.
3. Ha mulighet for å zoome til prisområder.
4. Visualisere rørledninger, noder med logistikk informasjon.
5. Vise flyt mellom prissoner.
6. Visualisere smart-mapping, metrokart.

Videre i dokumentet har vi tatt for oss ett og ett punkt hvor vi konkluderer hver nøkkelfunksjon, og oppsummerer med en tabell for økt oversikt.

7.1.1 Visualisere prissoner på kartgrunnlag

I de fleste tilfeller er en prissone et land. For disse tilfellene var det en triviell jobb å markere disse prissonene. Geografiske data (kartet) er lagret som en strukturert SVG fil. I programmeringsomgivelsen får man tilgang til hver prissone som et polygon, der man legger til funksjonalitet etter ønske. I dette tilfellet vil det være å fargelegge prissonen etter prisen på kraft, og vise prisinformasjon som tekst. Det var i starten problemer med å importere SVG kart inn i Visual Studio/Chart FX Maps på grunn av de strenge kravene for egendefinerte SVG kart. Etter at disse problemene var løst ble resultatene meget bra. Chart FX viser frem kartet på en pen måte, og manipuleringen av farger på kartet ble vellykket.

7.1.2 Visualisere tidsperspektiv på forskjellige prisområder

I en prissone vil kraftprisen oppdateres for hver time gjennom døgnet. På bakgrunn av dette ønsker Navita en mulighet for å kunne velge tidspunkt og få ut kraftprisen for den gitte timen. For å løse dette har vi delt inn døgnet i 24 timer, slik inndelingen er på Nord Pool Spot. Vi lagrer alle prisdata i en XML fil.

For å implementere tidsperspektivet bygget vi videre på løsningen fra forrige punkt. I tillegg til å legge til programkode ble det gjort endringer i oppbyggingen i XML filen for muligheten for tidsperspektiv.

Vi har laget et forløpsindikator der man kan velge time. Denne menyen henter riktig data fra XML filen basert på brukerens valg. Etter at riktig prisinformasjon har blitt hentet vil de aktuelle sonene fargelegges, den oppdaterte kraftprisen vil vises.

7.1.3 Mulighet for zoom

Zoom er nyttig for oversiktsbilder. Takket være Chart FX ble denne jobben med å implementere zoomfunksjonalitet triviell. Vi ser at også denne gang hjelper det oss å ha strukturert dataene. Det er en innebygd funksjon i Chart FX der man kan zoome inn til ønsket polygon. I dagens situasjon kan man zoome inn slik at man ser en tredjedel av Norge, eller hele Sverige dersom man velger dette som interessefelt. Vi mener at denne zoom-funksjonen vil være tilstrekkelig for Navitas fremtidige løsning. På små stater (polygoner) vil denne zoom-funksjonen gi enda bedre uttelling enn hos de store landene (prissoner).

7.1.4 Visualisere rørledninger, noder med logistikk informasjon

Dette punktet viste seg å bli problematisk, problemer med eksisterende datagrunnlag og Chart FX. Datagrunnlaget var laget for å illustrere grenser uten

referansepunkter til f. eks byer.

Dersom vi skulle gått videre med dette verktøyet måtte vi lagt inn informasjon direkte i kartgrunnlaget. Dette Navita mener er en dårlig løsning, de ønsker at kartet skal inneholde geografisk informasjon og ikke noe mer. Vi er enige med Navita på dette punkt. For å legge til en rørledning med geografisk korrekthet var vi nødt til å forsikre oss om at kartgrunnlaget er geografisk korrekt. I et forsøk på å gjøre kartgrunnlaget geografisk korrekt lagde vi en funksjon for å regne ut eksisterende koordinater til GPS koordinater. I teorien er dette en smart løsning. I praksis fungerte ikke denne teorien da SVG formatet ikke ville samarbeide med det nye koordinatsystemet som vi forsøkte å bruke. Andre faktorer som kartprojeksjon spiller også inn på sluttresultatet, hvor man er avhengig at man benytter seg av samme kartprojeksjon på kartgrunnlag og informasjonen man ønsker å vise på kartet.

Da vi sto igjen med et kart som vi ikke kunne garantere var geografisk korrekt, og ingen mulighet til å gjøre endringer i kartet, så vi oss nødt til å se etter andre verktøy for å lage applikasjonen i.

7.1.5 Flyt mellom prissoner

Det å visualisere flyten av kraft krever en dynamisk løsning. Vi skrev i forrige avsnitt at dette ikke er Chart FX sterke side, der vi var nødt til å tegne opp et statisk gassnettverk for i det hele tatt ha muligheten for å vise frem gassnettverket. Men å visualisere flyten er en mer dynamisk oppgave, ønsket vi å se hvilke muligheter andre verktøy kunne gi oss før vi fortsatte med Chart FX.

Da vi la opp planen for gjennomføring av prosjektet ble funksjonene prioritert i denne rekkefølgen som punktene er presentert i nå. Vi rangerte flyt mellom prissoner og metrokart funksjonaliteten som mer avansert enn de andre, og ble derfor planlagt til å utføres til slutt. Da vi ikke fikk til foregående oppgave med å visualisere gassnettverket ble alle våre ressurser vendt vekk fra Chart FX til Google Maps. Da vi mente at vi ikke fikk til en fullverdig løsning i Chart FX ønsket vi å utforske muligheten for å få en fullstendig løsning i en annen teknologi.

7.1.6 Visualisere smart-mapping, metrokart

Denne funksjonaliteten er en av de store utfordringene når det gjelder visualisering. Smart-mapping har blitt ønsket fordi det øker lesbarheten i kartet, men er ganske komplisert å implementere. Det kreves at linjer tegnes dynamisk, å gjøre denne jobben manuelt kan bli meget omfattende ettersom endringer vil skje i fremtiden. Vi vil igjen vise til lite fleksibilitet i Chart FX, der vi ønsker å legge til dynamikk i visualiseringen. Chart FX er best egnet til å vise statisk materiale.

7.1.7 Sammendrag skrivebordsløsning

Oppgavelisten for skrivebordsløsningen var på seks punkter. Av de seks punktene klarte vi å gjennomføre halvparten. De tre oppgavene vi klarte å gjennomføre var vi meget fornøyd med resultatet. Vi hadde strukturerte data som vi jobbet med, og fikk stor hjelp av dette under implementeringen. I teorien er vi enige i valget om å bruke Visual Studio og Chart FX Maps som benytter seg av SVG filer. Da vi hadde kommet skikkelig i gang med prosjektet følte vi selv at valgene som Navita hadde gjort med tanke på utviklingsverktøy og teknologi var riktig, og mente absolutt at vi skulle få laget et bra løsningsforslag. Etter hvert som utfordringene ble mer komplisert, og krevde mer av utviklingsverktøyet forandret vårt syn på Chart FX Maps seg. Det var en grunnleggende egenskap som vi trengte for å kunne fortsette arbeidet med Chart FX. Dette var muligheten for endre på data som allerede var fremvist. Vi følte at det var vanskelig å få en god fremdrift på prosjektet dersom vi skulle fortsatt å jobbe med Chart FX. Beslutningen om å utforske mulighetene i Google Maps ble tatt i samtale med Navita, for siden å ta opp arbeidet med Chart FX dersom vi anser det som fornuftig.

Vi arbeidet videre med Google Maps og fikk løst flere problemer i dette verktøyet enn Chart FX. Derfor ble resten av tiden i prosjektet dedikert til å jobbe med Google Maps, og vi prioriterte ikke ett nytt forsøk med Chart FX.

Vi har laget en tabell under for å vise med få ord hvilke oppgaver som ble gjennomført og ikke. Vi understreker at disse konklusjoner er tatt av forfatterne, med støtte fra Navita. Dersom Navita ikke var fornøyd med løsningsforslaget ble forslaget forsøkt forbedret.

Oppgave	Resultat
Visualisere prissoner på kartgrunnlag	God løsning, fornøyd med resultatet
Visualisere tidsperspektiv på forskjellige prisområder	God løsning, fornøyd med resultatet
Ha mulighet for å zoome til prisområder	God løsning, fornøyd med resultatet
Visualisere rørledninger, noder med logistikk informasjon	Ikke gjennomført, problemer med Chart FX
Vise flyt mellom prissoner	Ikke gjennomført, problemer med Chart FX
Visualisere smart-mapping, metro-kart	Ikke gjennomført, nedprioritert

Tabell 7.1: Oppsummering skrivebordsløsning

7.2 Webløsning

Vi ønsket å utforske mulighetene i Google Maps da vi vet at vi har et geografisk korrekt kartgrunnlag over hele verden. Dette kartet blir oppdatert av Google, så Navita slipper å tenke på vedlikehold av kartgrunnlaget. Valget falt på Google Maps over Microsoft Virtual Earth basert på tilgjengelig dokumentasjon og tilgang til andre prosjekter som benyttet seg av kartgrunnlaget til Google Maps.

Teknologi som vi har benyttet oss av for å utvikle webløsningen er Google Maps API v2.95, JavaScript, JSON og aspx.net. Funksjonskravene og utfordringene blir det samme som på skrivebordsløsningen.

1. Visualisere prissoner på kartgrunnlag.
2. Visualisere tidsperspektiv på forskjellige prisområder.
3. Ha mulighet for å zoome til prisområder.
4. Visualisere rørledninger, noder med logistikk informasjon.
5. Vise flyt mellom prissoner.
6. Visualisere smart-mapping, metrokart.

7.2.1 Visualisere prissoner på kartgrunnlag

Fordelen med Google Maps er at kartgrunnlaget er korrekt. Google Maps benytter GPS-koordinater (x,y) for å plassere objekter.

Utfordringen var å finne en måte å fargelegge de ulike prissonene. En måte å dele inn i prissoner er å benytte seg av overlays. Man kan tegne opp et omriss av f. eks Sverige, og fargelegge dette overlayet med ønsket farge. Vi har i våre demonstrasjonsløsninger vist Norden med slike overlays. Problemet igjen blir å dele inn Norges dynamiske prissoner som igjen krever at vi har tilgang til data for Norges kommunegrenser. Som en provisorisk løsning har vi delt inn i 3 statisk soner for demonstrasjonsformål. Vi lagrer prisinformasjon i en XML fil, mens geografisk informasjon lagres i en KML fil.

Overlayene hentes fra en KML fil som har lagret x,y punkter. Man er på forhånd nødt til å lage omriss av de sonene man ønsker å benytte. Til dette har vi laget en funksjon som gjør at man kan lage disse omrissene direkte i Google Maps. På denne måten slipper man unna med problemer som bruk av feil kartprojeksjon o.l. Disse omrissene kan lagres på sentralt og brukes som en overlay på ønsket sone.

Denne formen for å merke prissoner krever litt forarbeid, men dette er en jobb som må gjøres en gang for hver prissoner. Vi har laget en annen måte å dele inn i prissoner som er generell og ikke krever noe forarbeid i form av å lage polygoner manuelt. I stedet for å bruke overlays som fyller hele sonen, har vi laget et forslag som benytter seg av sirkler i prissonen. Denne formen for å skille prissoner gjør at man kan visualisere prissoner over hele verden uten å være avhengig å lage overlayet på forhånd. Funksjonen er meget enkel å

bruke for brukeren som gir navnet på landet som input, og funksjonen slår opp senterGPS-koordinatene til dette landet, og bruker informasjonen fra Geonames som bakgrunn for å plassere midtpunktet i sirkelen.

7.2.2 Visualisere tidsperspektiv på forskjellige prisområder

I motsetning til skrivebordsløsningen ble soneinndeling og tidsperspektiv laget i samme utviklingssteg. Vi lærte litt fra utviklingen av skrivebordsløsningen, slik at vi slapp å bruke unødvendig tid for å oppgradere fra statisk visning til visning med tidsperspektiv.

Tidsperspektivet ble løst ved å legge til en rullegardinliste der man velger hvilken time som er av interesse. Deretter tegnes Norden med farger etter prisnivå. Vi er fornøyd med denne løsningen, og denne kan utvides til flere prissoner etter ønske.

7.2.3 Mulighet for zoom

Google Maps har innebygd zoom, derfor er dette punktet egentlig unødvendig for webapplikasjonen. Men vi ønsker å presisere at zoom-funksjonen fortsatt har noen problemer med seg. Det var ingen problemer med de overlayene vi har laget, disse skalerte fint etter valgt zoom-nivå. Det er imidlertid ting man må tenke på når man skal visualisere informasjon som skal være zoombar. Vi må passe på at elementene, som f. eks ikoner og rørledninger blir oppdatert til en passende størrelse (detaljeringsgrad) i forhold til hvilket zoom-nivå man er på. Via Google Maps API kan man hente ut hvilket zoom-nivå bruker har valgt, slik at man kan tilpasse elementene sine deretter. Det er flere måter å skalere ikoner på, man kan enten bytte det fysiske bildet til en variant som er fysisk mindre, eller man kan stille selve størrelsen på ikonet. Vi har vist frem begge alternativene under demonstrasjoner til Navita. Som tidligere nevnt så er det ikke bare ikoner vi må tenke på ved endring av zoom. Rørledninger må bli tynnere/tykkere, og kanskje legges til eller fjernes. Dersom man zoomer inn til et nærområde ønsker man som regel å se detaljerte røرنettverk. Lokale rørledninger er ikke så interessant i et oversiktsbilde, der vi er avhengig av å ha de viktigste rørledningene med i visningen. En av grunnene til å implementere zoom er at brukeren selv skal kunne bestemme hvor detaljert visning de selv ønsker det.

7.2.4 Visualisere rørledninger, noder med logistikk info

Vi lar Google Maps tegne rørene selv, ved at vi gir start- og sluttkoordinater som argumenter til en funksjon som hjelper oss med dette. Den andre delen av oppgaven var å legge til noder med passende ikoner. Vi har laget noen eksempler der vi benytter oss av landbaserte og offshore oljeinstallasjoner og mulighetene for å legge til andre typer ikoner som koblingspunkter o.l. er en triviell jobb. I våre eksempler har vi fremvist informasjonen med en tooltip som vi kan legge

inn nødvendig informasjon. Google Maps tooltip støtter HTML-koding, derfor er mulighetene store for å kunne vise dynamisk informasjon.

7.2.5 Flyt mellom prissoner

Flyt mellom prissoner var en funksjon som er listet som det siste punktet på oppgavelisten. En av grunnene til dette er kompleksiteten til denne funksjonen, og denne funksjonen er avhengige at de grunnleggende funksjoner fungerer. Derfor ble denne funksjonen nedprioritert i fordel til de mer grunnleggende funksjonene. Planen var å implementere denne funksjonen til slutt, men på grunn av lite utviklingstid igjen valgte vi å forbedre de grunnleggende funksjonene.

7.2.6 Visualisere smart-mapping, metrokart

Det er en utfordring å lage kart med god lesbarhet, og vi har forsøkt å implementere denne funksjonaliteten med mindre hell. For at løsningen skal være brukbar på lang sikt kreves det at funksjonen tegner opp rørledningene dynamisk. Det var først i slutten av prosjekttiden av vi startet å se på denne funksjonaliteten. Vi har laget noen teoretiske eksempler på hvordan det kan se ut, men vi har ikke kommet fram til en løsning hvor systemet selv tegner opp rørledninger dynamisk i henhold til hvordan London Underground og andre metrokart har blitt bygd opp. På en annen side kan vi si at vi har en løsning som kan innfri metrokart løsningen. Dersom vi fjerner kartgrunnet, vil vi sitte igjen med kun rørledninger og deres noder. Dette kan være mer lesbart enn med kartgrunnet som underlag.

7.2.7 Sammendrag webløsning

I forhold til skrivebordsløsningen klarte vi å få løst flere del-oppgaver. Google Maps API er fleksibel nok til Navitas behov, og det finnes mye dokumentasjon om hvordan man bruker dette verktøyet. Det er helt klart en fordel med å bruke et eksisterende kartgrunnet som er vedlikeholdsfritt samtidig som det er geografisk korrekt. Vi føler i ettertid at det var et riktig valg å bruke Google Maps som verktøy for å utvikle denne løsningen. Google Maps er et satsingsområde for Google, noe som gjør at nye funksjoner blir utviklet etter hvert som gjør løsningen fremtidsrettet. Allerede fra starten av prosjekttiden til slutten fant vi nye funksjoner som vi kunne hatt nytte av. Det er spesielt en funksjon som kan hjelpe oss med flyt og logistikk info med tanke på rørledninger, Google Traffic Information. Denne funksjonen er beskrevet i seksjon 8.0.1.

Vi har laget en tabell under for å vise med få ord hvilke oppgaver som ble gjennomført og ikke.

Oppgave	Resultat
Visualisere prissoner på kartgrunnlag	God løsning, fornøyd med resultatet. Har flere løsningsforslag
Visualisere tidsperspektiv på forskjellige prisområder	God løsning, fornøyd med resultatet
Ha mulighet for å zoome til prisområder	God løsning, fornøyd med resultatet, detaljeringsgrad er tatt høyde for.
Visualisere rørledninger, noder med logistikk informasjon	God løsning, fornøyd med resultatet
Vise flyt mellom prissoner	Ikke gjennomført, nedprioritert, i teorien gjennomførbart*
Visualisere smart-mapping, metrokart	En prototyp har blitt laget, nedprioritert, i teorien gjennomførbart*

Tabell 7.2: *Viser til teori i seksjon 3.4 hvor ulike måter å gjøre å visualisere disse problemene har blitt beskrevet og diskutert.

7.3 Oppsummering

Tankegangen bak Chart FX Maps med SVG kart og C# som programmeringsspråk er bra. Disse teknologiene er moderne og har en fremtid foran seg. Vi gikk inn i prosjektet fri for fordommer og gjorde vårt beste. Da fremgangen stoppet på grunn av lite muligheter i Chart FX var vi nødt til å se i andre retninger for å finne et verktøy som kan gjøre jobben bedre. I tillegg til lite fleksibilitet i Chart FX hadde vi problemer med å stole på om kartgrunnlaget er korrekt. Mistanken om dette ble enda større når Chart FX egen supportavdeling ikke vet hvilken type kartprojeksjon kartet er basert på. Forsøk ble gjort på å gjøre kartet geografisk korrekt med GPS-koordinater, men en kombinasjon av SVG og Chart FX gjorde dette unødvendig komplisert. Vi mener at dersom vi skal ha en løsning som er geografisk korrekt er man nødt til å ha et kartgrunnlag som er korrekt og som man kan stole på. Dersom kartgrunnlaget er ukorrekt vil resten av applikasjonen være ukorrekt og en lang rekke med korreksjoner og kompensasjoner må gjøres for å holde geografiske korrekte elementer i nærhet med sin nye geografiske ukorrekte posisjon.

I prosessen med å finne et nytt verktøy å utvikle i falt valget på Google Maps, det kan i ettertid diskuteres om dette var et rett valg. Uansett så føler vi at det var rett valg da vi klarte å gjennomføre flere del-oppgaver enn det vi gjorde i Chart FX. Vi er overbeviste om at dersom vi hadde hatt mer tid til rådighet kunne vi gjennomført de resterende punkter i Google Maps. Dersom vi må velge hvilket verktøy vi ville benyttet for videre fremgang for applikasjonen er svaret klart; Google Maps har vært lettere å jobbe med, gitt oss bedre resultater, og har et kartgrunnlag som er meget bra. Dersom Navitas ønske om skrivebordsapplikasjon er viktigere enn disse parameterne, anbefaler vi å se på et annet verktøy enn Chart FX Maps. Vi føler oss sikre på at finnes muligheter

for å lage denne applikasjonen som en skrivebordsapplikasjon i Visual Studio. En av de store utfordringene blir å finne et kartgrunnlag som er godt nok.

I oppgavebeskrivelsen skriver Navita at de primært ønsker en skrivebordsapplikasjon, men er interessert i å se på mulighetene i internettbaserte kartprogrammer. Denne oppgaven har vært et utforskningsprosjekt der Navita har gitt oss en liste med utfordringer som har forsøkt å løse i de forskjellige verktøyene. Helt fra første stund har Navita gitt uttrykk for at det ikke er noe garanti for at alle oppgavene kan løses i de rammene vi har hatt. Oppgaven vår har hele tiden å teste muligheten i teknologien, slik at Navita kan gjøre sine vurderinger om valgt teknologi er bra nok.

Med denne problemstillingen i bakhodet føler vi at har gjort en god jobb med dette prosjektet. Vi har løst flesteparten av Navitas oppgaver, samtidig som vi har funnet andre problemer som Navita nå er klar over. Disse problemene er viktig å være klar over dersom Navita bestemmer seg for å ta steget videre med applikasjonen. Hvis vi hadde vært i Navitas sko så mener jeg at flere problemer som har blitt oppdaget er til hjelp senere i utviklingsprosessen. Flere av disse problemene har blitt diskutert og laget forslag til løsninger på. Noen problemer har vært vanskeligere å løse, mens noen få har vært umulige å løse.

Vi har skrevet et fremtidig arbeid kapittel der vi tar for oss hva som er neste steg i prosessen. I dette kapittelet tar for oss nye funksjoner som kan være hjelpfulle, og hvilke funksjoner som bør være med i løsningen.

Oppgave	Skrivebordsløsning	Webløsning
Visualisere prissoner på kartgrunnlag	Fullført, god løsning	Fullført, god løsning
Visualisere tidsperspektiv på forskjellige prisområder	Fullført, god løsning	Fullført, god løsning
Ha mulighet for å zoome til prisområder	Fullført, god løsning	Fullført, god løsning
Visualisere rørledninger, noder med logistikk informasjon	Ikke fullført, Chart FX problemer	Fullført, god løsning
Vise flyt mellom prissoner	Ikke fullført, Chart FX problemer	Ikke fullført, har teoretiske løsninger
Visualisere smartmapping, metrokart	Ikke fullført, nedprioritert	Delvis gjennomført

Tabell 7.3: Oppsummering av hvilke arbeidsoppgaver som ble fullført for skrivebordsløsning og webløsning.

Kapittel 8

Fremtidig Arbeid

Utviklingen av en skrivebordsløsning basert på Chart FX vil ikke anbefales, til det blir ikke Chart FX dynamisk nok. På bakgrunn av arbeidet med løsningene som er utviklet ville vi fokusert det fremtidige arbeidet på en webløsning som benytter Google Maps. Dette kapittelet vil derfor ta for seg hvilken funksjonalitet man kan implementere i webløsningen, og mulige fremtidige tekniske løsninger som kan være interessante å benytte seg av. Webløsningen er delt opp i fire individuelle løsninger og en naturlig utvikling vil være å sette disse sammen til en løsning. Vi har laget en illustrasjon, figur 8.1, som beskriver hvordan vi ser for oss en slik løsning.

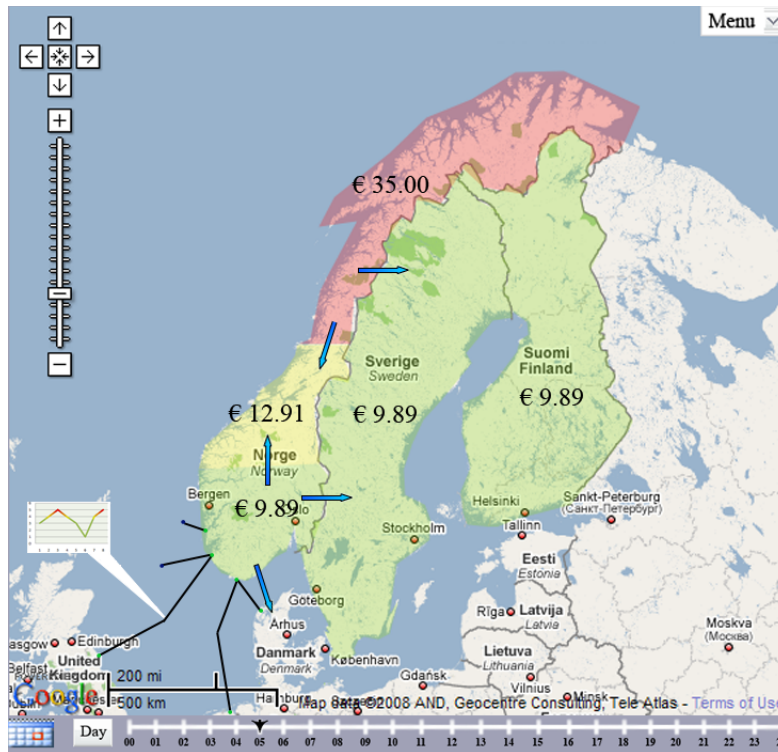
De ulike elementene som vi har sett for oss er:

Kalender Gir mulighet for å søke til en spesifikk dato for så å visualisere tilhørende kraftpris.

Tidslinje Tidslinjen vil ta over for rullegardinlisten og vil inneholde flere funksjoner enn kun å bla gjennom timer. Ved å velge visningstype vil man kunne bla seg gjennom timer, dager, uker osv.

Prisvisning Dagens Google Maps API inneholder ikke funksjonalitet for å visualisere tekst direkte på kartet. I fremtiden vil dette bli implementert, men selv i dag kan man benytte seg av tredjepartsløsninger for å løse dette. En av disse er laget av Michael Purvis og gir mulighet for å opprette tekst på markeringspunkter [21]. Purvis har er nå ansatt i Google Maps teamet og ryktene blant diverse diskusjonsforaer på internett hinter til at dette vil bli implementert nå som Michael Purvis har blitt engasjert i Google-teamet.

Flytgrafer Mulighetene er mange når det kommer til å vise flytmønstre og en av de vi synes var interessant var å benytte informasjonsboksene fullt ut. Boksene benytter standard HTML og kan på den måten inneholde alt fra ren tekst til bilder og videoer. Dette gir mulighet for å visualisere flytmønstre ved bruk av grafer slik illustrasjonen viser.



Figur 8.1: Illustrasjon av fremtidig webløsning

Meny Denne vil gi mulighet for å velge hva slags informasjon man ønsker å visualisere, noen brukergrupper har ikke behov for å se logistikk informasjon og andre vil muligens kun ha behov for logistikk informasjon. Det å ha mulighet for selv å tilpasse løsningen etter egne krav vil øke brukervennligheten betraktelig.

8.0.1 Google Traffic Information

Teknologien er konstant under utvikling og Google lanserte på tampen av prosjektperioden et nytt hjelpemiddel til Google Maps. Dette kalles trafikk informasjon og viser veiarbeid og fargekoder veiene etter hvor mye trafikk som er på veiene til enhver tid. Dette kan være nyttig for Navita å se videre på.

Mulighetene for å lage et oversiktlig bilde av logistikken, da med tanke på flytmønstre, er i dag ikke helt trivielt å få til i Google Maps. Når det kommer til kraftledninger som bærer over lengre strekninger, kan flyten variere på deler av strekningen. For å visualisere dette på en god måte, kunne det være ønskelig å fargekode flytnivået og på den måten kunne man på en enkel måte se forskjellene i flytnivået og også oppdage flaskehals. I dagens Google Maps API er det kun mulig å sette en farge per ledning, skal man få til denne løsningen må man



Figur 8.2: Google Maps Traffic Information

dele opp ledningene og fargelegge disse hver for seg. Google Maps APIen er i stadig utvikling og en interessant løsning på dette problemet kan være i deres trafikk analyse system, som vist i figur 8.2. Her kan man tydelig se at veiene blir merket i forhold til flytnivå, og et slikt system kan være en mulig løsning for å visualisere flytmønstrene i kraftledninger. Per dags dato er ikke disse funksjonene frigitt i Google Maps, men muligens er dette grunnet at Google vil teste ut systemet før det frigis til allmennheten. Trafikk analyse systemet inneholder også lettleste symboler som inneholder informasjon på spesifikke punkter, slik som trafikkulykker, veiarbeid etc. Dette kan også være interessant å benytte seg av i kraftledningssystemet, da det kan inneholde informasjon om stengte gassledninger, utbedring av ledningsnett, ulykker som har rammet ledningsnettet etc.

Bibliografi

- [1] Andre goncalves, chart fx technical support team.
<http://mysupport.softwarefx.com/Help/SupportTool/LatestReply.aspx?IID=e7d89cb1-fa1d-4a8c-a55d-93994d641b2e> (Lesedato: 26.mai 2008).
- [2] Bret taylor, product manager google.
<http://groups.google.com/group/Google-Maps-API/msg/8222b18e7921f6e6> (Lesedato: 26.mai 2008).
- [3] Fakta om navita. <http://www.navita.com/?CatID=1169> (Lesedato: 26.mai 2008).
- [4] Google maps mania blog. <http://GoogleMapsmania.blogspot.com> (Lesedato 3.juni 2008).
- [5] Googles terms of service. <http://www.google.com/accounts/TOS> (Lesedato 2.juni 2008).
- [6] Halden centre of expertise, hjemmeside.
http://www.forskningsparken.no/HCoE-NO/ARENA_prosjekt/ (Lesedato 3.juni 2008).
- [7] Hjemmeside for geonames. <http://www.geonames.org/about.html> (Lesedato: 2.juni 2008).
- [8] Nord pool spot glossary. <http://www.nordpoolspot.com/Glossary/> (Lesedato: 26.mai 2008).
- [9] Nordpool area price. <http://www.nordpoolspot.com/reports/areaprice/> (Lesedato: 26.mai 2008).
- [10] Nordpools historie. <http://www.nordpoolspot.com/about/History/> (Lesedato: 26.mai 2008).
- [11] Offisiell hjemmeside for software fx, produsent av chart fx maps.
<http://eu.softwarefx.com/sfxNetProducts/ChartFX/> (Lesedato: 26.mai 2008).

- [12] Telegeography hjemmeside. <http://www.telegeography.com> (Lesedato 2.juni 2008).
- [13] Telegeography's traffic flow map. http://mappa.mundi.net/maps/maps_014/#ref_5 (Lesedato: 2.juni 2008).
- [14] T. A.Keahey. *The generalized detail-in-context problem. In Proceedings of the IEEE Symposium on Information Visualization, 1998.* IEEE Computer Society pp 44-51.
- [15] Ryan Asleson and Nathaniel T. Schutta. *Foundations of Ajax.* Apress publishing, 2006.
- [16] Joachim Böttger, Martin Preiser, Michael Balzer, and Oliver Deussen. Detail-in-context visualization for satellite imagery. *Eurographics 2008*, 27.
- [17] Bugayevskiy, Lev, and John Snyder. *Map Projections: A Reference Manual.* Taylor and Francis, 1995.
- [18] Leo M. Hurvich. *Color Vision.* Sunderland, Massachusetts, 1981.
- [19] T. A. Keahey and E. L. Robertson. *Techniques for non-linear magnification transformations. In Proceedings of the IEEE Symposium on Information Visualization, 1996.* IEEE Computer Society pp 38-45.
- [20] Mican Lacher. *Teach yourself SVG in 24 hours.* Sams Publishing, 2002.
- [21] Andre Lewis, Michael Purvis, Jeffrey Sambells, and Cameron Turner. *Beginning Google Maps Applications with Rails and Ajax.* Apress Publishing, 2007.
- [22] Matthew MacDonald and Mario Szusztá. *Pro ASP.NET 2.0 in C# 2005.* Apress Publishing, 2005.
- [23] McDonnell and Porter. *Introduction to Map Projections.* Landmark Enterprises, 1991.
- [24] Torgeir Nakken. Gir penger, prestisje og jobber (lesedato 3.juni 2008). 2008. URL til artikkel: <http://www.ha-halden.no/artikkel.asp?Artid=169209>.
- [25] Keith V. Nesbitt. *Getting to more abstract places using the Metro map metaphor.* School of Information Technology, Charles Sturt University, Bathurst, Australia, 2004.
- [26] International Hydrographic Organization. *General Bathymetric Chart of the Oceans.* Ottawa, Canada, 5th edition, 5.06, 1984.

- [27] Jennifer Preece, Yvonne Rogers, and Helen Sharp. *Interaction Design: beyond human-computer interaction*. John Wiley and Sons inc., New York, 2002.
- [28] Håkon Tolsby. Forelesningsnotater. GrensesnittDesign, HiØ 2006.
- [29] Edward Tufte. Offisiell hjemmeside.
http://www.edwardtufte.com/bboard/q-and-a-fetch-msg?msg_id=00005W&topic_id=1 (Lesedato: 26.mai 2008).
- [30] Edward Tufte. *The Visual Display of Quantitative Information*. Graphic Press, Box 430, Cheshire, Connecticut 06410, 1983.
- [31] Edward Tufte. *Envisioning Information*. Graphic Press, Box 430, Cheshire, Connecticut 06410, 1990.
- [32] Edward Tufte. *Visual Explanations*. Graphic Press, Box 430, Cheshire, Connecticut 06410, 1997.
- [33] Danny Goodman with Michael Morrison. *JavaScript Bible 5th Edition*. Wiley Publishing inc, www.wiley.com, 2004.
- [34] Michael J. Young. *XML Step by Step Second edition*. Microsoft press, 2002.
- [35] Øyvind Lie. Halden får verdensledende kraftutdanning. *Kraftjournalen*, 2008. URL til artikkel:
http://www.arenaprogrammet.no/Arena_fs/arena%20milj%C3%B8vennliggihandel.doc.

Figurer

1.1	Illustrasjon av Nord Pool Spot	8
2.1	XML-skjema basis.	16
3.1	Nordpool spot illustrasjon	21
3.2	Ruteplan for Czechoslovakia Air Transport Company 1933	22
3.3	Modifisering av generelle prissoner løsningen	23
3.4	Trafikkflyt kart for telekommunikasjon	24
3.5	London Underground kart anno 1933	26
3.6	London Underground kart anno 2008	26
3.7	Ulike varianter av metrokart	28
3.8	Forsøk på geografisk korrekte løsninger	30
3.9	Forsøk på geografisk korrekt webløsning	31
3.10	Illustrasjon av prisskala der vi ser prisskalaen til skrivebordsløsningen øverst og skalaen til webløsningen nederst. (Tallene referer til hvilken pris som gjelder til hver farge.)	32
3.11	Kontinentalvisning	34
3.12	Detaljeringsgrad	34
3.13	Ikoner	35
4.1	Illustrasjon av Chart FX diagram fremvisning.	38
4.2	Illustrasjon av Chart FX Maps fremvisning.	38
4.3	Illustrasjon av Google Maps	40
4.4	Illustrasjon av overlay	45
4.5	Manuelle overlays	45
4.6	Illustrasjon av generell overlay	46
4.7	Projeksjonsfeil	47
4.8	Sylindriske projeksjoner	48
6.1	Kommuneinndelt SVG	59
6.2	Skrivebordsløsning hovedbilder	60
6.3	Skisse av skrivebordsløsningen	60
6.4	Demonstrasjon av zoom funksjon i skrivebordsløsningen	61
6.5	Skisse av hvordan man oppretter prissoner	62
6.6	Verktøy for inndeling av prissoner	63

6.7	Visualisere prissoner	64
6.8	Skisse av generelle prissoner	65
6.9	XSD av Generelle prissoner	65
6.10	Demonstrasjon av generelle prissoner	66
6.11	Demonstrasjon av generelle prissoner prisinformasjon	66
6.12	XSD som brukes for visualisering av rørledninger	67
6.13	Visualisere rørledninger demonstrasjon	68
6.14	Informasjon knyttet til elementene	68
8.1	Illustrasjon av fremtidig webløsning	80
8.2	Google Maps Traffic Information	81

Tabeller

7.1	Oppsummering skrivebordsløsning	73
7.2	Oppsummering webløsning	77
7.3	Oppsummering av begge løsninger	78

Tillegg A

Delivery

Dette kapittelet inneholder det tekniske dokumentet som ble overlevert til Navita. Dokumentet er skrevet på engelsk, noe som var ønsket av Navita, og tilbakemeldingen vi fikk fra Navita var positiv: "Den tekniske beskrivelsen var veldig bra, så det var ikke noe problem å få dette til å kjøre hos oss. Koden virker grei å forstå, så vi planlegger ikke med noen oppfølging på dette."(Svein Olsen).

A.1 Desktop application

The desktop application is developed using Visual Studio.net 2005 and the use of Chart FX 7.0 with the maps extension.

A.1.1 Nordpoolspot

The project is using 3 external files.

Map_NorNed.svg This is the custom map that is used by the application.

load.txt The load.txt is a xml file which is the default loaded price list.

Test.txt This is the test xml file that got all the prices for the different regions.

It is important that the Map_NorNed.svg is located in the Chart FX installation directory, the path where the SVG file should be located is:

```
"Chart FX for Visual Studio 2005\Extensions\Maps\Library\Regions"
```

This is because the map is loaded directly from this location in the development process, after compilation it could be located in the program directory. When the program executes, the data from load.txt will be imported to the map chart table and the data from Test.txt would be loaded into the Chart2 table. When the user changes the time using the track bar, the corresponding data in chart2 would be loaded into chart1, which is the map chart. The time is displayed in a label left of the track bar.

A.2 Web application

The web pages using the Google Maps API need a genuine key which is mapped to the web server domain. To attain a key go to <http://code.google.com/apis/maps/>. This requires a Google Account since the API key is connected to a Google Account. The key variable is loaded in the main web page on every web application and looks like this:

```
...maps.google.com/maps?file=api&v=2.95&key=insert_key_here" type=...
```

These projects have been tested in the Mozilla Firefox web browser and it is not guaranteed that they will work in other browsers such as Internet Explorer, Opera, Safari etc.

A.2.1 Circleoverlay

The web site includes 4 files:

index.html Main page, includes most of the JavaScript.

jsr_class.js Jason script to make dynamic request tags.

locations.xml The data file which contains the different locations and prices.

mapStyle.css Cascading Style Sheets

This is a standard web application which uses html and JavaScript. The main page uses the `body=onload` function to load the map functions in the JavaScript. There are two drop-down menus on the right side, the one on the top changes the hour and the other changes the opacity on the overlays. When there is a change in either of them, it will execute either `changehour(which)` or `changeopacity(which)`. These are the script entry points.

A.2.2 Gaspipes

The web site includes 4 files:

index.html Main page.

gaspipes.js The Javascript file.

gaspipes_multilines.xml The data file which contains the different locations with additional information.

mapStyle.css Cascading Style Sheets

This is a standard html and JavaScript page. In addition to the files described above there is also necessary to include the icon picture files. The index file uses the `body=onload` function to execute the JavaScript. The data from the xml file is loaded in a series of functions to first load the pipelines and then the markers (icons) is loaded. The icon picture is based on the `<Type>` field in the xml.

A.2.3 Nordpoolspot

The web site includes 4 files:

Default.aspx Main page.

javascript.js The Javascript file.

price.xml The data file which contains the different locations price data.

mapStyle.css Cascading Style Sheets

This is a ASP.net project, but it could easily be transformed to a standard html site, since it is not using any ASP.netfunctions. As the other projects this uses the `body=onload` function to load the map. The dropdown menus on the right side are similar to those in the Circleoverlay project. The overlay coordinates are coded directly into the JavaScript since this project is made for testing purposes, it would probably be more useful to store them externally, such as a database etc.

A.2.4 Overlaymaking

Default.aspx Main page.

Default.aspx.cs ASP code file which contains the save function.

javascript.js The JavaScript file.

mapStyle.css Cascading Style Sheets

overlay.xml The exported overlay file.

This ASP.NET project is using a server side function which could not be implemented in JavaScript, because JavaScript doesn't allow saving files on the web server. This project is user operated, the first mouse click on the map will place a start marker, and the following clicks will render a line from the previous point to the point clicked. To store the overlay press the "Make listbutton, which makes a list of geographical coordinates in the textbox, then this could either be copied directly or saved on the web server using the Savebutton. This stores the data within a KML file, "overlay.kml", and it could be retrieved from the server to be used in Google Earth or the Nordpoolspot"web page.

Tillegg B

Demonstrasjon av webløsning

Oversikriftene referer til Tillegg A som ble overlevert til Navita. Vi valgte å holde overleveringsdokumentet uendret, derfor har vi lagt ut linker til demonstrasjon av de ulike webløsningene i dette vedlegget. Demoene er testet med Internet Explorer og Firefox. Demoene er tilgjengelig en begrenset periode.

B.1 Circleoverlay

<http://www.locco.info/Master/geonames2.html>

B.2 Gaspipes

<http://donau.hiof.no/anderslo/final/Gaspipes/index.html>

B.3 Nordpoolspot

<http://donau.hiof.no/anderslo/Final/nordpoolspot/>

B.4 Overlaymaking

<http://donau.hiof.no/anderslo/final/Overlaymaking/>