DNMI DET NORSKE METEOROLOGISKE INSTITUTT

klima

DATABASE-PROSJEKTET I KLIMAAVDELINGEN. STATUS PR. 23.12.92

P. Øgland, K.A. Iden, P.O. Kjensli, S.L. Lystad,
M. Moe, B. Nordin, Å.M. Vidal, T. Aasen.

RAPPORT NR. 53/92 KLIMA

DNMI - RAPPORT

DET NORSKE METEOROLOGISKE INSTITUTT
POSTBOKS 43 BLINDERN 0313 OSLO 3

TELEFON: (02) 96 30 00

ISBN

RAPPORT NR.
53/92 KLIMA

DATO
23.12.1992

TITTEL

DATABASE-PROSJEKTET I KLIMAAVDELINGEN.
PROSJEKTPROSEDYRER, STANDARDER, DATASTRUKTURER
OG DATAFLYT. STATUS PR. 23.12.92

UTARBEIDET AV

P. Øgland, K. Iden, P. O. Kjensli, S. L. Lystad,
M. Moe, B. Nordin, Å. M. Vidal, T. Aasen.

OPPDRAGSGIVER

DNMI

SAMMENDRAG

Rapporten inneholder status for databaseprosjektet pr. 23.12.92. Dette innebærer en forenklet fremstilling av hovedresultater presentert i DNMI-KLIMA rapportserien 1992.

Prosjekthistorikken gjennomgåes, og teknikker for sikring av kvalitet internt i prosjektet drøftes.

Videre oppsummeres filosofien og konsekvensene ved valg av datastrukturer for geofysiske og administrative data. Dataflyten skisseres, og prinsipper for systemutvikling eksemplifiseres gjennom midlertidige resultater fra arbeid med kontroll- og applikasjonssystemer.

Prinsipper og resultater fra innlasting av historisk data omtales. Videre arbeid skisseres.

UNDERSKRIFT

Petter Øgland
FORSKER

Bjørn Aune
FAGSJEF

I N N H O L D

1. PROSJEKTREDEGJØRELSE

Databaseprosjektet - Klimaavdelingen ble startet opp ved årsskiftet 1989/1990.

Prosjekthistorikken så langt består av

- Utarbeidelse av kravspesifikasjon for innkjøp av database og databasemaskin (1990), se [1].

- Forstudium, spesielt med henblikk på utarbeidelse av en informasjonsmodell og oversikt over flagging og kontroller (1991), se [2].

- Utarbeiding og testing av ulike datastrukturer, etablering av valgt datastruktur, utarbeiding av standarder for systemutvikling og kvalitetsstyring for prosjektarbeid (1992), sml. [3], [4], [5] og [6].

Prosjektmedlemmer pr. 23.12.92 er Margareth Moe (prosjektleder), Knut A. Iden, Bjørn Nordin, Per Ove Kjensli, Sofus L. Lystad, Åse Moen Vidal, Petter Øgland, Tom Aasen.

1.1 Fremdrift

Den planmessige fremdriften av arbeidet i 1992 har vært styrt av en prosjektplan [7] med påfølgende revisjoner, for siste revisjon se [8]. Planen har stykket opp det totale arbeidet i åtte underprosjekter.

Prosjekt 1: Kursing og opplæring
Prosjekt 2: Utarbeiding og testing av ulike datastrukturer
Prosjekt 3: Etablering av valgt datastruktur på Typhoon
Prosjekt 4: Standarder for systemutvikling
Prosjekt 5: Kvalitetssikring for prosjektarbeid
Prosjekt 6: Kontrollert overgang til det nye systemet
Prosjekt 7: Utvikling av rutiner og applikasjoner
Prosjekt 8: Videre utvikling av datasystemet

Pr. i dag er prosjektene 1 til 5 gjennomført, og prosjektene 6 og 7 i gang. Prosjekt 8 vil bli initiert når de øvrige prosjekter er avsluttet.

Prosjekt 1 "kursing og opplæring" var organisasjonsmessig forskjellig fra de senere prosjekter, og besto av kartlegging av opplæringsbehov for utarbeidelse av en systematisk opplæringsplan sam realisering av denne ved enkelte prosjektmedlemmers deltagen i eksterne kurs med temaer som UNIX-operativsystem, databaseadministrasjon, databaser og SQL, SQL*Plus, PL/SQL, 3GL/Oracle, SQL*Forms. Kompetansen har så blitt overført til databasegruppen gjennom forelesninger og kollokvievirksomhet. Prosjektet ble påbegynt 01.11.91. Siste formelle bidrag til prosjektet var en intern forelesningsrekke over programmeringsspråket C september/oktober 1992.

Prosjekt 2 "Utarbeiding og testing av ulike datastrukturer" ble påbegynt 01.11.91, men grunnet prosjektets store omfang ble det stykket opp i enheter; Klassifisering av meteorologiske parametre, Utarbeiding av ulike datastrukturer til testing, Overføring av data og datastrukturer til Typhoon, Oppretting av testmiljø, Lagring av systeminformasjon, Kartlegging av bruk av data samt hvilken bruk som er tidskritisk. Prosjekt 2 ble avsluttet 05.11.92 i form av DNMI-rapport 42/92 KLIMA [4] utarbeidet av P. Øgland (prosjektleder), K. Iden, P. O. Kjensli, S. Kristiansen, S. L. Lystad, M. Moe, B. Nordin, Å. M. Vidal, T. Aasen.

Prosjekt 3 "Etablering av valgt datastruktur på Typhoon" baserte seg på hovedresultatene i prosjekt 2 og kunne således ikke påbegynnes før prosjekt 2 hadde presentable resultater. En foreløpig prosjektplan ble utarbeidet 22.04.92, mens den produktive delen av arbeidet ble påbegynt 03.09.92. Prosjekt 3 ble avsluttet 07.10.92 og dokumentert i DNMI-rapport 40/92 KLIMA [3] utarbeidet av Å. M. Vidal (prosjektleder), S. L. Lystad, P. Øgland, M. Moe.

Prosjekt 4 "Standarder for systemutvikling" har på basis av generelle erfaringer, spesielle erfaringer i forbindelse med databasearbeidet i forstudien og prosjektene 1,2 og 3 lagt opp en standard for systemutvikling for den videre gjennomføring av prosjektet. Arbeidet ble påbegynt 08.11.91 og ble avsluttet 17.11.92 med DNMI-rapport 44/92 KLIMA [5] utarbeidet av B. Nordin (prosjektleder), M. Moe, K. A. Iden, P. O. Kjensli.

Prosjekt 5 "Kvalitetsstyring for prosjektarbeid" ble påbegynt 08.11.91. Prosjektet har utarbeidet en kvalitetssikringsplan for databaseprosjektet i tråd med Norsk Standard for kvalitetssystemer NS-ISO 9000 serien, og ble avsluttet med DNMI-rapport 45/92 KLIMA [6]17.11.92 utarbeidet av P. O. Kjensli (prosjektleder) og M. Moe.

Prosjektene 6 og 7 viser seg å inneholde såpass mange momenter at det er hensiktsmessig å splitte dem opp i mindre enheter. Pr. i dag arbeides det med temaene rutiner og applikasjoner for innlasting av data, rutiner og applikasjoner for kontroll av data, rutiner og applikasjoner for uthenting og behandling av data samt temaet datasikkerhet. Uthenting og behandling av data synes å være det mest omfattende blant disse.

1.2 Status

Denne rapporten har til hensikt å redegjøre for status i databaseprosjektet pr. 23.12.92, og tar utgangspunkt i rapportene [3] til [6] pluss noen foreløpige resultater og erfaringer i etterkant av disse.

I seksjon 2 oppsummeres noen av prinsippene benyttet for kvalitetssikring av prosjektet. Disse angir premissene for resterende arbeid og er ment å sannsynliggjøre effektiv og sikker fremdrift.

Med seksjon 3 vurderes de geofysiske forutsetningene for lagring av data. Hvordan lagre geofysisk og administrative data på en mest mulig hensiktsmessig måte er et fundamentalproblem som ble forsøkt løst i [4] og som rekapituleres her i korte trekk.

Dataflyt og prinsipper for program- og applikasjons-utvikling i databasen er tema for seksjon 4. Her prøver man å anskueliggjøre standard for systemutvikling [5] i lys av

dataflytarbeider dokumentert i [3] samt foreløpige resultater pågående virksomhet i databasegruppen.

Seksjon 5 redegjør for teori og praksis benyttet for fysisk lagring av data på Typhoon. Seksjonen er en oppsummering av nøkkelresultatene fra [3].

Rapporten er skrevet for alle som måtte ha interesse av et generelt innblikk i utviklingen av klimadatabasen. Ønsker man mer teknisk informasjon og grundigere redegjørelse for de enkelte resultater henvises man til referanselisten.

2. KVALITETSSIKRING

Hensikten med å benytte kvalitetssikring er å sikre at databasen gir riktig informasjon på en funksjonell måte i henhold til gitte spesifikasjoner. Prosjektet valgte å innføre en kvalitetsplan for databaseprosjektet basert på Norsk Standard for Kvalitetssystemer, NS-ISO 9000 serien.

Med utgangspunkt i definisjoner for hva som menes med kvalitet og kvalitetssikring (seksjon 2.1) fremsettes to sett med verktøy for å sikre kvaliteten i prosjektet:

- En kvalitetssikringshåndbok for prosjektet som definerer organiseringen av prosjektet, spesifiserer prosjektstyringsverktøy og setter krav til dokumentstyring.

- Intern kvalitetsrevisjon basert på skjema med selvreviderende funksjon.

2.1 Definisjoner

Med **kvalitet** menes en egenskap ved et produkt som sier noe om dets evne til å oppfylle fastsatte krav eller angitt behov.

En konkret anskueliggjørelse av kvalitet og kvalitetsbeslektede begreper er statistisk kvalitetskontroll. I statistisk kvalitetskontroll for en produksjonsprosess bruker man gjerne det relative antall defekter som et mål på kvaliteten. Dersom man ved stikkprøver kan registrere avvik fra gitt kvalitet antar man at man har kontroll over kvaliteten.

Med **kvalitetssikring** menes en fastsatt prosedyre for å sikre at et produkt eller en tjeneste får den ønskede kvalitet.

Med **kvalitetsstyring** menes mengden av de konkrete teknikker og aktiviteter som benyttes for å ha kontroll over kvaliteten.

2.2 Kvalitetsplan for databaseprosjektet

Siktemålet for kvalitetsstyringen defineres gjennom **kvalitetsplanen** som er det dokument som beskriver den særegne praksis for å oppnå kvalitet for et produkt eller et prosjekt. I den aktive styring benytter man seg så av **kvalitetsrevisjoner**, dvs. systematisk og uavhengig undersøkelse om kvalitetsplanen følges og er hensiktsmessig. For å kontrollere om kvalitetsplanen følges og er hensiktsmessig måler man **avvik**, dvs. mangel på oppfyllelse av spesifiserte krav, og **feil**, dvs. mangel på oppfyllelse av krav for tiltenkt bruk, altså bruk som ikke var forutsatt under kravspesifikasjonen.

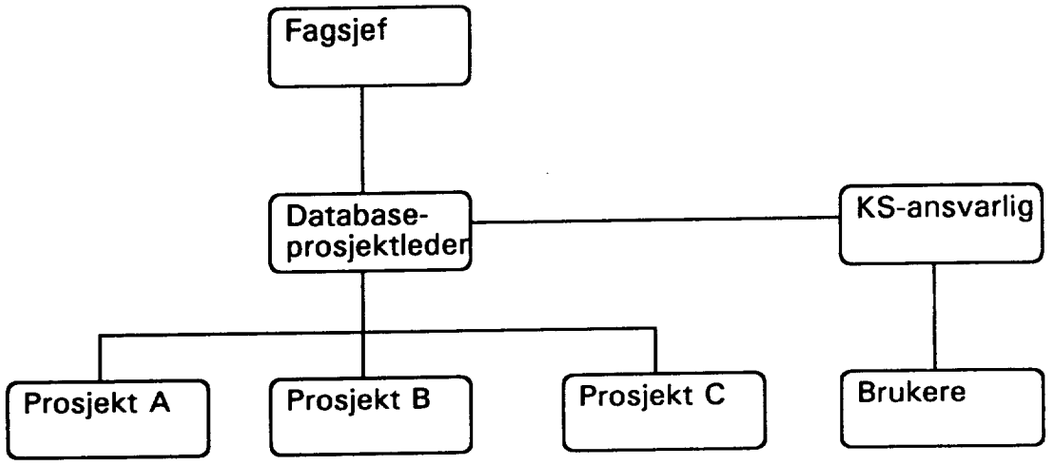
Kvalitetsplanen for databasearbeidet baserer seg på to faktorer, organisasjon og prosedyre. Ansvar og oppgaver er fordelt på en slik måte at det ikke skal kunne foreligge usikkerhet.

Til enhver tid løper 3-4 delprosjekter parallelt med egen bemanning og prosjektledelse. Delprosjektenes virksomhet er styrt av prosjektplaner og regelmessig møtevirksomhet hvor

korttids strategier legges opp. Delprosjektledelsen rapporterer til databaseprosjektleder som igjen rapporterer til fagsjef.

Side om side med prosjektets hierarkiske produksjonsstruktur finnes det en flat kvalitetssikringsstruktur. Kvalitetssikringspersonale, representert ved en kvalitetssikringsansvarlig (KS-ansvarlig), har en myndighet og et ansvar som svarer til prosjektleder for å sikre at prosjektet leverer den kvalitet som er stipulert.

KS-ansvarlig kan betraktes som ansvarlig overfor brukere og oppdragsgivere med hensyn på at prosjektet styres kvalitetsmessig i henhold til gitte normer. N. Langgård er KS-ansvarlig for klimadatabaseprosjektet.



Figur 2.1 Kvalitetssystem for databaseprosjektet

I tillegg til organisasjonsstruktur inneholder kvalitetsplanen en rekke konkrete metoder som benyttes i de aktiviteter som gjelder prosjektet. Metodene fordeler seg på planlegging og gjennomføring.

For å sette et mål på den ønskede kvalitet er det utarbeidet en kvalitetssikringshåndbok som beskriver prosjektstruktur og arbeidsmetoder.

For å sikre en effektiv gjennomføring benyttes kvalitetsrevisjon. I praksis betyr dette utfylling av kontrollskjemaer som skal dokumentere at KS-håndboken følges.

2.3 Kvalitetssikringshåndbok for databaseprosjektet

Det er utarbeidet en kvalitetssikringshåndbok for databaseprosjektet. Boken inneholder oversikt over

- Personell
- Verktøy til prosjektstyring
- Faseinndeling av prosjektene og prosedyrer for gjennomføring av fasene
- Dokumentasjonsstyring
- Revisjonsregler

Prosjektets personell består av prosjektleder, delprosjektledere, prosjektdeltagere og kvalitetssikringsansvarlig. Hvert delprosjekt samt hovedprosjektet fører et kvalitetssikringsskjema for prosjektet.

Møtevirksomhet føres regelmessig, styrt av prosjektplan og dokumentert gjennom referater. Oppfølging av prosjektene gjøres ved Gantt-diagrammer (grafisk angivelse av prosjektets aktivitetsforløp som funksjon av tid) og redegjørelse i prosjektledermøte. Det føres timelister.

Hvert enkelt prosjekt er stykket opp i faser. **Forstudiefasen** består av kartlegging av behov og undersøkelse av mulig løsninger. **Spesifikasjonsfasen** er en systematisk gjennomgang av problemet hvor beskrivelser av behov, krav og løsninger dokumenteres. **Konstruksjons- og realiseringsfase** består av analyse av løsningsmuligheter, skisse av disse og implementasjon. Man oppdaterer en avviksrapport som redegjør for avvik mellom konstruksjon og spesifikasjon. **Verifikasjonsfasen** baserer seg på gjennomgang av en testplan. Testing utføres og godkjennes av KS-ansvarlig.

For nærmere detaljer angående kvalitetssikring og adminstrasjon av prosjektet se [6] og [9].

3. UTARBEIDING AV DATASTRUKTURER

Kravene til effektivitet og sikkerhet er vitale under arbeidet med ordning av data i tabellstrukturer. Erfaringer dokumentert i [4] viste at valg av hensiktsmessig tabellstruktur var et ikke-trivielt problem på grunn av de store datamengder som skal lagres.

For å sikre seg at de hyppigst brukte parametre og stasjonsgrupper skal få optimal aksess baserte det videre arbeidet seg på en kartlegging av databruk ved Klimaavdelingen. Kartleggingen besto av en spørreundersøkelse og en statistisk analyse av forespørsler.

Fra spørreundersøkelsen fikk man inntrykk av at de mest populære geofysiske parametre er temperatur, vind (hastighet og retning) og nedbør. De mest brukte konstellasjonene av en eller flere parametre, stasjoner og tidspunkter så ut til å være som listet nedenfor, i rangert rekkefølge.

1. Flere parametre, en stasjon, flere tidspunkter.
2. Flere parametre, en stasjon, et tidspunkt.
3. Flere parameter, flere stasjoner, flere tidspunkter.
4. En parameter, en stasjon, flere tidspunkter.
5. En parameter, flere stasjoner, flere tidpunkter.
6. En parametre, flere stasjoner, et tidspunkt.
7. Flere parametre, flere stasjoner, et tidspunkt.
8. En parameter, en stasjon, et tidspunkt.

Forespørselsundersøkelsen besto av andelsestimater basert på et utplukk på 187 fra avdelingens 2663 skriftlige besvarelser 1991 (sml. DNMI årsberetning 1991).

Stasjonstypeestimatene antydet at de hyppigst forekomne forespørselene gikk mot Værstasjoner (68%), uspesifiserte stasjoner (13%), Vær/Nedbør-stasjoner kombinert (11%), og Nedbørstasjoner (8%).

De mest populære former for utskrifter var: Måneds- og årsmidler (35%), Daglige verdier (33%), Normaler (13%), Fuktighetsfordelinger (14%).

Det ble også gjort statistikk over hvor gamle data forespørselene rettet seg mot. I 31% av tilfellene var man interessert i over 10 år gamle observasjoner. I 20% av forespørselene var alderen uspesifisert, men i kun 4% av tilfellene ønsket man data for inneværende måned.

3.1 Lagringsstruktur for geofysisk data

Geofysiske data må lagres på to måter for å tilfredstille kvalitetskrav og en fornuftig arbeidsmetodikk, nemlig arbeidslager (AL) for midlertidig lagring og kontroll, og hovedlager (HL) for endelig lagring. Lagrene må ha forskjellig struktur med tanke på hva slags oppgaver som skal løses. I arbeidslageret vil alle stasjoner og parametertyper være representert på en slik måte at all innkommet data, uansett hvor urimelig, skal kunne representeres, og med en flaggstruktur for å holde orden på kontrollhistorikken. Formater benyttet i AL vil ligge så tett opp mot dagbøker og andre input-formater som mulig.

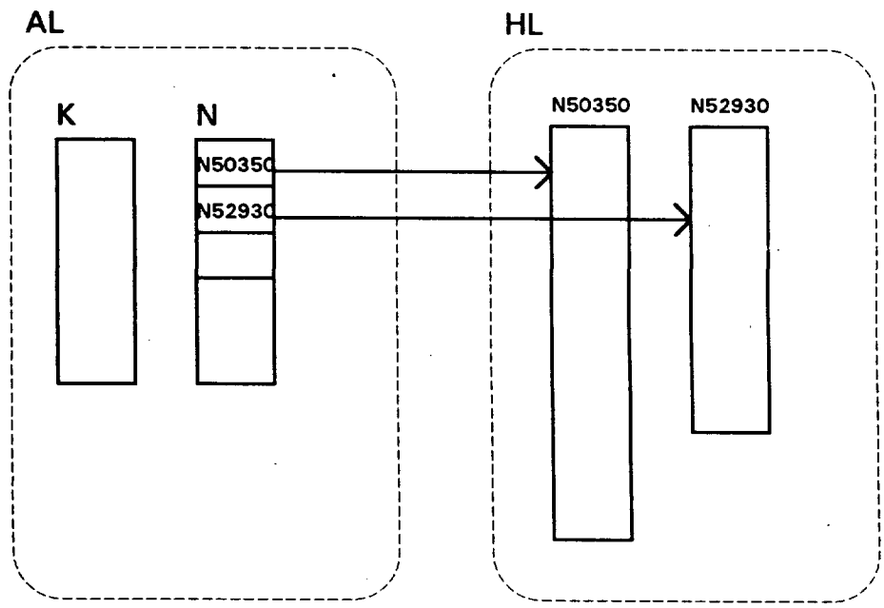
Hovedlageret inneholder samme data-informasjon, men lagret på en slik måte at det blir enklest mulig for sluttbruker å dra nytte av lageret. Dette betyr at lageret vil inneholde langt mindre grad av kodede verdier, og være mer uniformert med hensyn til enheter. Kontrollflagg benyttes i HL på samme måte som i AL, men reduseres i omfang til å inkludere flagging for informasjon om en værparameter for et gitt tidspunkt er interpolert/observert og om den er kontrollert (dvs. om registreringen ansees for å være offisielt korrekt).

Med utgangspunkt i behovsestimatene skissert ovenfor, kombinert med hva som er praktisk gjennomførbart, forsøkte man å komme frem til en fornuftig tabellstruktur for de geofysiske data.

Studier av parametergruppevis og områdevis (stasjonstypevis) lagring er utført. Stasjonstypevis klassifisering er definert som følger:

K - Vær, Linke- og Fordampningsstasjoner
N - Nedbørstasjoner
P - Plumatic stasjoner
M - Maritime stasjoner
A - Automatstasjoner (Edas, Scanmatic, Campbell, DNMI spec.)
E - E-data
L - Aanderaa
R - Radiosondestasjoner
F - Flyværstasjoner (Metar)

Stasjonstypevis inndeling med sammenhengende begrensede tidsrekker ser ut til å være det mest gunstige for AL, mens HL bør benytte stasjonsvis inndeling med komplette tidsrekker.



Figur 3.1 Stasjonstypevis og stasjonsvis tabellstruktur i henholdsvis arbeids- og hovedlager

Tabellene assosiert med hver stasjonsgruppe får navn etter stasjonsgruppen og et femsifret tall svarende til stasjonsnummeret. Nummeret vil være en utvidelse av eksisterende stasjonsnummer med en etterfølgende 0. Ved å inkludere bokstavkoden i tabellnavnet gjør man det enklere å behandle stasjoner som skifter stasjonsgruppestatus ved et gitt tidspunkt.

En detaljert beskrivelse av tabellstrukturen med parameternavn, flagg, enheter, koder og beskrivelser er å finne i [4].

3.2 Lagringsstruktur for administrativ data

I [2] introduserte man et informasjonslager for lagring av diverse administrative data. Grunntankene fra denne modellen er videreutviklet og modellen er i kontinuerlig vekst ettersom behov melder seg i forbindelse med programutvikling for innlasting, kontrollering, utlasting og viderebehandling.

Av hensyn til de forskjellige behandlinger databasen kan bli utsatt for er informasjonslageret delt opp i fire overlappende enheter.

1. Stasjons-informasjon
2. Parameter-informasjon
3. Instrument-informasjon
4. System-informasjon

Stasjonslageret inneholder informasjon om hvor stasjonen er lokalisert, hvilke stasjonsholdere som fins, hvilke vassdrag stasjonen er knyttet til, hvem som er observatør ved stasjonen, i hvilken tidsrytme målinger blir tatt og annet.

Parameterlageret inneholder informasjon som benyttes for etablering av nye tabeller innenfor gitte stasjonsgrupper, ekstremer, frekvenser og annen statistikk knyttet til parametre og kombinasjoner av parametre, og andre forhold som har med de enkelte parametre å gjøre. Tabellene vil bli benyttet blant annet til innlastingskontroll og homogenisering.

Instrumentlageret inneholder informasjon om instrumenter og inspeksjoner.

Systemlageret inneholder informasjon om hvordan databasen er bygget opp i henhold til prinsipper beskrevet i neste seksjon.

4. DATAFLYT OG PRINSIPPER FOR SYSTEMUTVIKLING

Denne seksjonen tar for seg noen av ideene for hvordan konstruere en mest mulig fleksibel og sikker database med tilhørende applikasjoner. Ideene er formulert som en standard [5], men vil i denne seksjonen kun gi pekepinn om hvordan programmer og dokumentasjon i forbindelse med databasen er tenkt realisert.

Rutiner og applikasjoner i forbindelse med utviklingen av klimadatabasen må struktureres på en sikker og effektiv måte. For å sikre at programutviklingen blir mest mulig samkjørt har prosjektet spesifisert en rekke forhold. Dette inkluderer skjermbilde-maler, navigerings- og behandlingsregler i skjermbilder i forbindelse med sluttbrukerprogrammering, lagringsstrukturer for forskjellige typer data og programvare, krav til benyttelse av programmeringskonvensjoner og versjonsstyring for utvikling av de enkelte programmer.

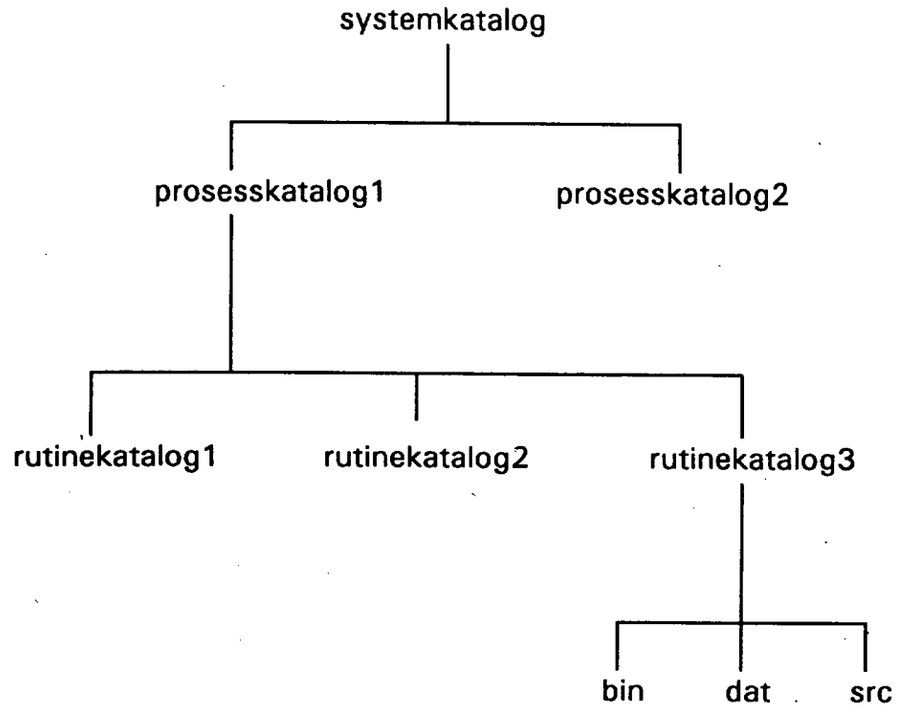
Målet er å oppnå et enhetlig og oversiktlig system, noe som vil være av betydning både for brukere og vedlikeholdere av systemet.

4.1 Utviklingsmiljø

Standard konvensjoner for programmering og lagring av data i UNIX-miljøer benyttes. For å sikre funksjonell oversikt har prosjektet basert seg på en konseptuell tredeling av programvare.

Et **system** er tenkt som den mest generelle form for programvare, og lar seg best visualisere for sluttbrukerne i form av en meny. Et undersystem kalles pr. konvensjon for en **modul**. Et system er bygget opp av **prosesser**, dvs. styringsjobber skrevet i satsvis modus med andre ord script/batch/mode avhengig av hva slags operativsystemkonvensjon som benyttes. Prosessene på sin side består av en rekke kall til forskjellige typer **rutiner**, dvs. programmer i C, FORTRAN eller SQL-varianter.

De forskjellige programvaretypene er lagret i katalogstrukturer under UNIX hvor de generelle systemene ligger øverst, prosessene ligger på et midlere nivå, mens programkoden ligger i de nederste katalogene. Rutinekatalogene forefinnes alltid i grupper på tre med navnene bin, dat, og src og inneholder henholdsvis binærkode, datakode og kildekode ("source code") for det aktuelle program.



Figur 4.1 Lagringsstruktur for programvare

For ytterligere å kunne skille mellom forskjellige programvare innenfor en gitt katalog benyttes navnekonvensjoner av typen SSS_ppp for å vise at systemet SSS benytter en prosess ppp lokalisert i katalogen ppp. Tilsvarende konvensjon for prosesser og rutiner, dvs. prosessnavnet qqq_rrr i katalogen qqq indikerer at prosessen benytter en rutine rrr fra katalogen rrr.

Oversikt og informasjon bør, så langt det er mulig, vedlikeholdes automatisk. Blant annet vil system-lageret omtalt i seksjon 2 være nyttig. Operativsystemets verktøy for programorganisering bør brukes og kan danne grunnlaget for systemoversikten. Aktuell verktøy under UNIX innebefatter make, ar og sccs.

Alle kataloger av typen src skal inneholde en ny katalog SCCS og et script Makefile. Se [4] eller on-line-manualene på Silicon Graphics for nærmere beskrivelse.

4.2 Strukturering av dataflyt

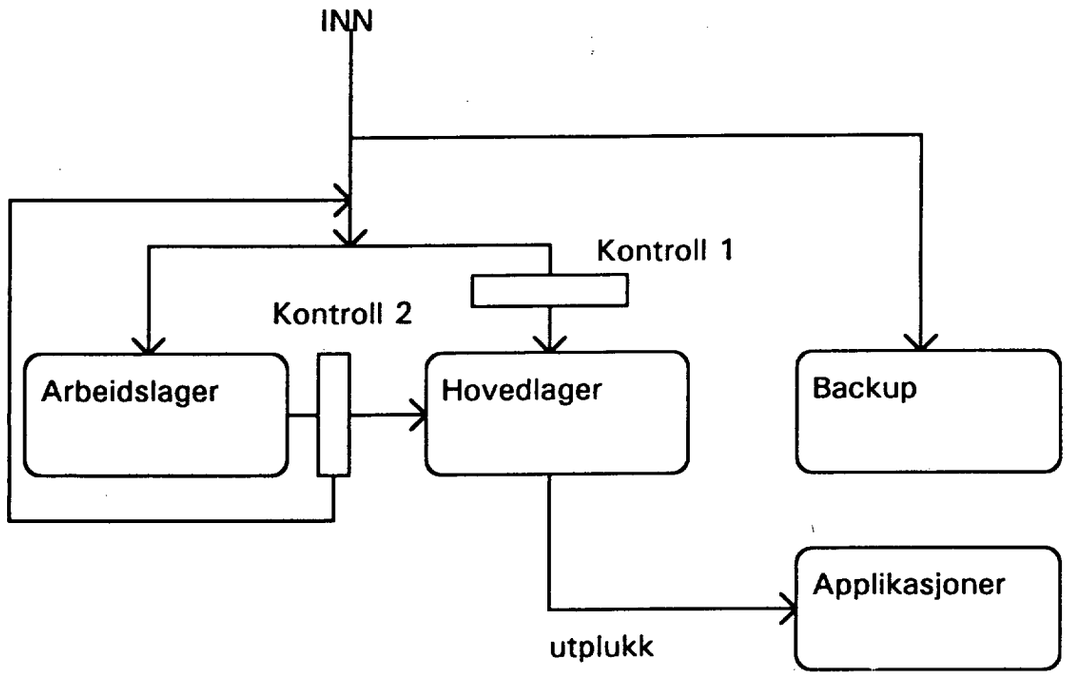
Dataflyten fra utenverdenen til databasen (Typhoon) går gjennom diverse forgreninger. All data kommer inn gjennom en felles port. Den første forgrening består så i at rådata kan tappes over på eksterne medier i form av backup. Denne ekstra sikkerheten kan være gunstig dersom det skulle oppstå feil i kontrollrutinene eller lagringsstrukturene.

Den virkelige dataflyten renner så videre til et nytt forgreningspunkt hvor data duplikeres og sendes simultant i to retninger. Den ene flyten går mer eller mindre direkte til arbeidslageret, mens den andre flyten går gjennom et kontrollfilter som hindrer at innlysende gale verdier havner i hovedlageret. Kontrollfilteret generer logg-filer som melder fra hvilke feil som er registrert.

Når data fra en innlastingssesjon er ferdig innlastet i AL startes kontrollfilter 2. Dette filteret tester data mot diverse tester, avhengig av mengde og type data i AL og HL. Dersom data ikke godkjennes av testen genereres en gjetning (interpolasjon) som sendes tilbake for ny innlasting. Gjetninger kan være både manuelle og automatiske. Ny innlasting og nye kontroller gjennomføres.

Dersom data skulle slippe gjennom kontrollen fra AL til HL settes et flagg "kontrollert" i HL, og et flagg "interpolert" dersom verdien er behandlet.

Utplukk til bruk for applikasjoner gjøres fra HL.



Figur 4.2 Dataflyt

Kontroll-filterene er fasedelte, dvs. de inneholder flere trinn med kontroller hvor en kontroll på trinn *k* ikke vil bli startet før kontrollen på trinn *k*-1 er avsluttet. Dette svarer til hvordan kontrollrutinene funger pr. i dag, dvs. månedskontrollen er avhengig av at ukeskontrollene er gjort. Den første fasen i kontroll 2 gjøres så snart data er kommet inn i systemet. Senere kontroller kan f. eks. være avhengig av at flere stasjoner fra et geografisk område er inne simultant.

Den lokale flyten innenfor et filter, i forbindelse med utplukk eller applikasjoner er pålagt å følge visse standardiserte krav. Systemet skal splittes opp i enheter bestående av styreprogram, formateringsprogram, autorisasjonsprogram, feilkontroll og styring av prosesslogg og feillogg.

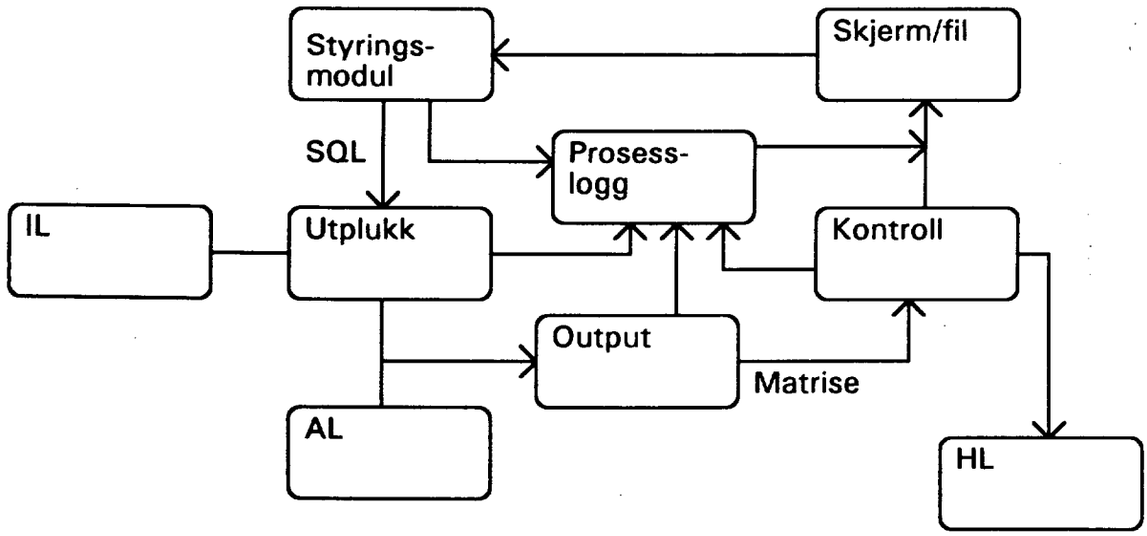
Kontroll-systemet

Kontrollfilter 1 er et innlastingsfilter som kun har som oppgave å hindre uønsket data å slippe inn i HL. Kontrollfilter 2 derimot, som inneholder mer interessante former for kontroller, er å betrakte som et system, og må inneholde systemkomponentene beskrevet ovenfor.

Systemet trigges med en styringsmodul. Denne har til hensikt å sikre at de rette utplukk og de rette kontroller utføres. Styringsmodulen generer SQL-kode som oversendes utplukksalgoritmen.

Utplukket kan nyttegjøre seg av administrative data fra informasjonslageret (IL). Resultatet av utplukket oversendes så kontroll-modulen via en output-modul. Outputmodulen formaterer Oracle-svaret til en matrise for å sikre standardisert input til sjekk-rutinene inneholdt i kontroll-modulen.

I kontrollmodulen vil en passende rutine bli startet, avhengig av hvilken fase og hvilke flagg som er satt av styringsmodulen. Kontrollmodulen setter inn interpolasjonsverdi og interpolasjons-flagg i HL dersom en test ikke blir bestått, og kontroll-flagget i HL ellers. Flagg i AL oppdateres for å dokumentere hvilke kontroller som er gjennomført.

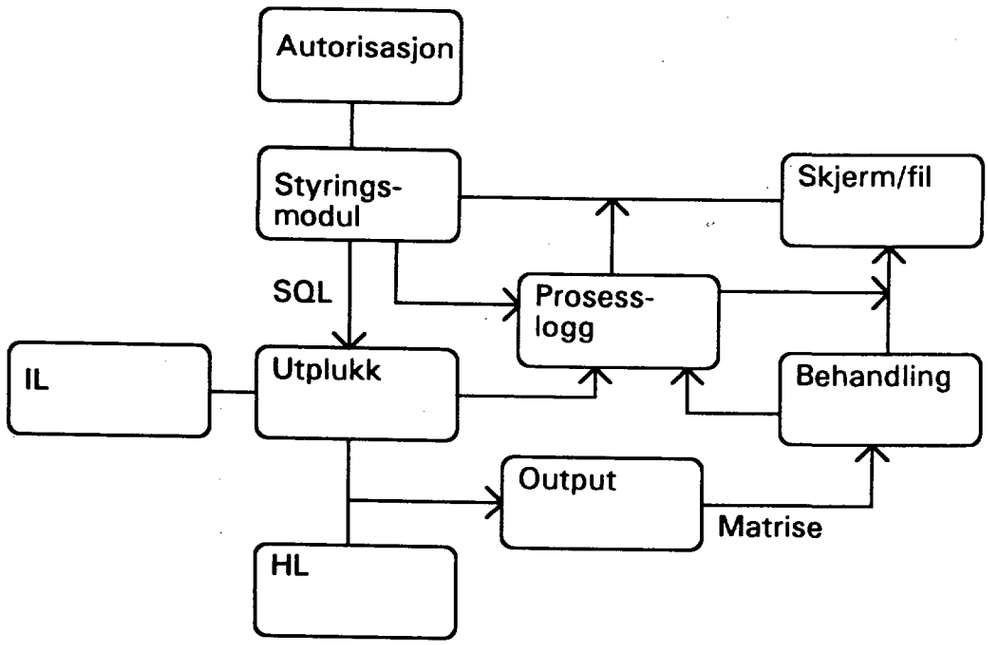


Figur 4.3 Kontrollsystem

Applikasjons-systemet

Applikasjonssytemet er organisert etter et tilsvarende generelt mønster. En styringsmodul kommuniserer med autorisasjonsrutinene for å sjekke rettigheter. Input registreres og danner grunnlaget for utplukk. Setningene går gjennom en utplukksrutine hvor utplukkssetningene optimaliseres. Ved hjelp av kommunikasjon mot IL gjøres det ønskede utplukk, som så formateres gjennom outputkanalen til matriser eller tabeller.

Tabellinformasjonen flyter videre inn til standardiserte behandlingsrutiner, og gjennom disse ut til skjerm/fil og styringsmodulen.



Figur 4.4 Applikasjons-system

Flyten for både kontroll- og applikasjonssystemene er programmert i UNIX. Rutinene i hver modul er under utvikling.

4.3 Programmering

For å sikre at programmeringen utføres mest mulig modulært og sikkert må man stille krav både til programmeringskonvensjoner og programdokumentasjon.

I prinsippet ønsker man oversiktlig programmering slik at behovet for kommentarfelter blir minimalt. De kommentarer som blir ført skal beskrive logikk og innhold.

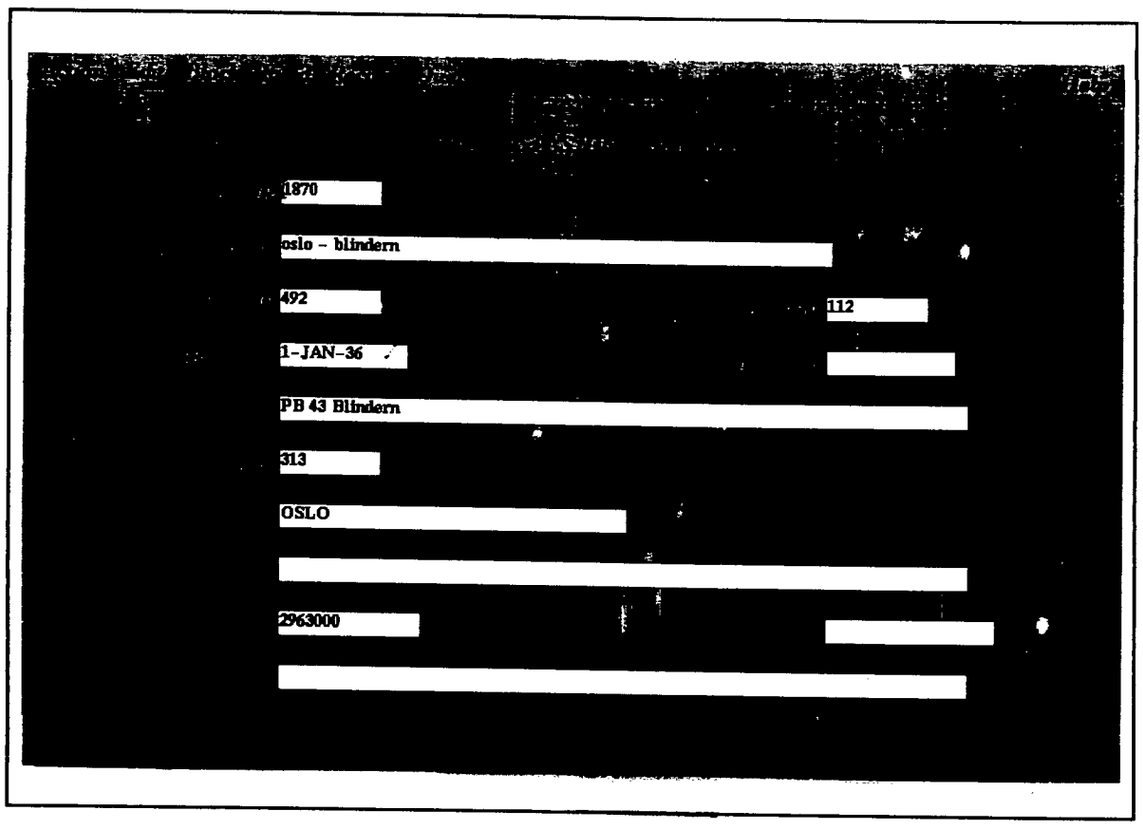
I et flerutvikler-miljø er også kravene til versjonsstyring og generell programutviklingsinformasjon for de enkelte kildefiler nødvendige. UNIX-systemet SCCS vil ivareta hovedkravene. Andre krav, f. eks. spesifikasjon av Input/Output, og hvordan systemene skal spesifiseres og dokumenteres er redegjort for i [5], kap. 3.5 og kap. 7.

Generell programutvikling utføres hovedsakelig i FORTRAN, C og SQL. Man har forsøkt å legge seg på en mest mulig lærebok-orientert standard, hvor kravene til oversikt, modularitet og effektivitet aksentueres. Retningslinjer for Fortran og C følger samme grunnfilosofien hvor kompakt programkode, hyppig bruk av hjelpebiblioteker, prosedyrer og bunting av variable utgjør noen av grunnpilarene.

I valget mellom hva som skal gjøres i C og hva som skal gjøres i SQL vil problemets kompleksitet og krav til effektivitet spille inn. Forøvrig prøver man å utnytte shell-programmering under UNIX i størst mulig grad da dette erfaringsmessig er den beste og raskeste måten å oppnå maksimal sikkerhet og komfort.

4.4 Standard for brukergrensesnitt og dokumentasjon

Figuren nedenfor viser et typisk skjermbilde benyttet for innlasting av stasjonsinformasjon.



Figur 4.5 Typisk skjermbilde

Skjermbildene skal være opererbare på en måte som ikke medfører unødig ergometrisk slitasje for brukeren, og de skal være generelt enkle å bruke.

Navigasjon mellom forskjellige skjermbilder kan foregå på flere måter, men prinsipielt vil man benytte seg av menyer.

Mer utførlig beskrivelse av standarder for brukergrensesnitt foreligger i [5].

5. ETABLERING AV DATASTRUKTUR PÅ TYPHOON

Denne seksjonen tar for seg de praktiske sidene ved etablering av de strukturer som er referert tidligere.

Seksjon 5.1 behandler integritetsregler for å øke den logiske sikkerheten i databasen.

Seksjon 5.2 behandler segmentering for å øke den fysiske effektivitet og sikkerhet.

Seksjon 5.3 refererer eksperimenter med indeksering i de geofysiske lagre.

Seksjon 5.4 ser på innlastingsrutiner og eksperimenter.

5.1 Utnyttelse av integritetsregler i databasen

Integritetsregler skal sikre at databasen er konsistent, at den ikke inneholder ulovlige verdier og at det ikke forekommer redundans.

Regler som ivaretar ovenforstående kan alltid implementeres manuelt i applikasjoner, men ORACLE gir mulighet for å legge disse direkte inn sammen med dataene i setninger benyttet for å lage tabeller (CREATE) eller modifikasjon av disse (ALTER setningene). Enhver operasjon av data (INSERT, UPDATE, DELETE), enten interaktivt eller fra egenutviklede applikasjoner, vil da bli underkastet det samme regelverk uten at man behøver å tenke på dette selv.

Oracle versjon 6 gir mulighet for å spesifisere en rekke integritetsregler i CREATE / ALTER setningene, men for de fleste av disse blir det ikke tatt hensyn til disse spesifiserte reglene i versjon 6 av databasen. I versjon 7 vil imidlertid reglene bli fulgt fullt ut.

SQL*FORMS versjon 3.0 kan i visse tilfeller (ved default skjermbildeopprettelse) benytte integritetsreglene fra v.6 av databasen.

Integritetsreglene er:

- **NOT NULL**
- **PRIMARY KEY**
- **FOREIGN KEY**
- **UNIQUE**
- **CHECK**
- **DEFAULT**

NOT NULL krever at kolonnen er fylt ut med noe (verdi). Denne regelen blir utnyttet i versjon 6.. PRIMARY KEY er en spesifisering av primærnøkkelen, dvs. kolonner som entydig definerer raden og kan brukes som søkebetingelser for data. FOREIGN KEY er en henvisning til primærnøkkelen i en annen tabell og krever at kolonnene er veldefinerte og inneholder data i den fremmede tabellen. UNIQUE krever at kolonnene er entydige innenfor tabellen. Dette kan delvis implementeres i v. 6 ved opprettelse av UNIQUE

INDEX, men forutsetter altså opprettelse av indeks - noe som ikke alltid er like fornuftig. CHECK spesifiserer en liste av verdier, et verdiområde og eksakt verdi fra en oracle-funksjon. DEFAULT brukes til å gi elementene i kolonnen en standard verdi.

Integritetsreglene vil bli benyttet både for geofysiske og administrative lagringsstrukturer og implementeres etterhvert som det blir nødvendig. Integritetsreglene kan slettes i ALTER setningen med DROP CONSTRAINT. Navnet på denne vil alltid kunne finnes i ORACLE DICTIONARY.

5.2 Segmenteringsprinsipper for databasen

Tabeller, indekser, og hjelpestrukturer for midlertidig lagring plasseres av Oracle i såkalte TABLESPACE (tabellrom). Tabellrommene kan fordeles over flere disker, og ved å strukturere tabeller indekser og hjelpetabeller innen tabellrommene skal det være mulig til en viss grad å styre effektivitet og sikkerhet.

Tabellrom

For å få en best mulig yteevne må diskaksessene spres mot flere disker for samtidig Input/Output (IO). Dette oppnås ved å fordele tabellrommene som inneholder IL, HL og tilhørende indekser på forskjellige disker.

Et tabellrom kan spenne over 1 til flere disker. Dersom et tabellrom som spenner over flere disker skulle bryte sammen, blir hele tabellrommet satt ut av drift inntil den ødelagte disken blir erstattet. Tabellrommet er enhet for backup. Hvis et tabellrom spenner over flere disker, kreves det like mange backup-disker tilgjengelig samtidig og jobben tar forholdsvis lengre tid.

Av disse grunnene ser det ut til at geofysiske data må sorteres stasjonstypevis i egne tabellrom. Indekser og data for samme stasjon skal spres på forskjellige disker. Dette vil redusere samtidig IO. IL-tabellrommet bør ikke ligge på samme disk som HL eller dens indekser. Samme regler gjelder også for AL og dens indekser sålangt det er mulig med hensyn til antall disker tilgjengelig. HL må imidlertid prioriteres med hensyn på yteevne.

Forventet vekst i databasen

Man kan avsette plass i databasen etter et planlagt mønster for hvor mye man tror den vil vokse. Avsetter man såpass plass at databasen kan vokse vil man hindre ukontrollert oppsplitting av tabeller på flere disker. En ukontrollert oppsplitting (fragmentering) vil øke IO og dermed responstiden. Skal man forhindre fragmentering kombinert med begrenset plass må man benytte reorganisering. Reorganisering innen en disk er enkelt, men reorganisering av hele databasen er en omfattende oppgave.

Som en start setter databasegruppen av plass (INTIAL EXTENT) til å inneholde historisk data pluss data for 3 år fremover. NEXT EXTENT settes til å romme datamengde for 1 år fremover.

Da man forventer at tilveksten i databasen vil være av tilnærmet lineær orden settes den prosentvise veksten (PCTINCREASE) til 0%.

For valg av verdier for andre lagringsparametre se [3].

5.3 Indeksering

Indekseringen på AL og HL må følge forskjelligè prinsipper ettersom de skal benyttes på forskjellige måte. På dette stadium har det vært naturlig med en skissemessig betraktning av indekseringsmuligheter både på AL og HL. Den endelige beslutningen på hvordan indekser skal velges for de respektive lagre avhenger av de løpende prosjekter som gjør særstudier av kontroll- og applikasjons-systemer. På et foreløpig tidspunkt kan man gjøre flere mer eller mindre velfunderte gjetninger.

Diverse eksperimenter med indeksering er gjennomført for studier av AL. I AL forventer man at kun primærnøkkelen, dvs. stasjonsnr (STNR), år (AAR), måned (MND), dag (DAG) og time (TIM), vil inngå i indekseringen.

Et slikt begrenset utvalg skyldes at store indekser er tunge å oppdatere, og de ser også ut til å være mindre effektive i bruk.

En test på en 448953 rader lang tabell M2222 ble gjennomført med utplukkssetningen "SELECT STNR, AAR, MND, DAG, TIM, TT FROM M2222 WHERE STNR=1870 AND AAR=1980 AND MND=3 AND DAG=1". Oppslaget returnerte tre rader.

Et eksperiment besto av 326 forskjellige indeks-tester på utplukk fra M2222. I eksperimentet noterte man seg fire forhold, nemlig hvor lang tid det tok å genere en indeksstruktur, hvor lang tid det første oppslaget tok, og forventet oppslagstid hvis man gjentok utplukket noen ganger.

Grunnen til at man ikke nøyde seg med et førstegangsoppslag er at utplukkshastigheten kan forbedres radikalt dersom all data som skal hentes befinner seg i lokalhukommelsen (sml. [4] kapittel 5.2).

Ved å sortere med hensyn på beste kombinasjon av første oppslag og senere oppslag slo følgende indekskombinasjoner best ut:

Indeks-struktur	Umiddelbar responstid	Forventet responstid
(aar,mnd,stnr,tim,dag)	0.3000	0.0458
(aar,dag,mnd,stnr,tim)	0.3166	0.0417
(aar,stnr,mnd,dag,tim)	0.3333	0.0417

Forskjellen mellom responstiden for disse utplukkene besto i et par hundredels sekunder. Den forventede responstiden er en forbedring på ca. 84%. Tiden det tok å kreere slike indekser var ca. 800 sekunder (13 min.).

Blant andre interessante indekskombinasjoner fant man (aar,stnr,dag,mnd) med umiddelbar respons 0.4166 sek. og forventet respons 0.0542 sek. Indeksen inneholdt ikke timeparameteren, som forøvrig ikke var med i betingelsen for SELECT setningen.

Kombinasjonen (aar,dag) slo også godt an med umiddelbar respons på 0.6500 sek. og forventet respons på 0.3417 sek. I dette tilfellet var forventningen en forbedring på ca. 47%, mens tiden det tok å kreere indeksen var 259 sekunder (4.3 min).

For videre arbeid med AL brukes imidlertid (aar,mnd,stnr,tim,dag) som presumptivt beste indeks.

Indeksering av HL må knyttes mot aktuelle applikasjoner. En mulig løsning er å benytte (aar,mnd,tim,dag) sammen noen nøkkelparametre (temperatur, vindhastighet, vindretning, nedbør), eller, hvis denne indekskombinasjonen viser seg å være for stor, benytte (aar, dag) sammen med et lite utvalg geofysiske parametre.

5.4 Innlasting av historisk data

Det er gjort diverse tester med innlasting av historisk data. Et innlastingssystem er utviklet som laster inn data parallelt til AL og HL. På veien til HL passerer datastrømmen et enkelt kontrollfilter, sml. figur 4.1.

Innlastingssystemet baserer seg på kontinuerlig kommunikasjon med IL. Kommunikasjonen består i kontroller om stasjonen er registrert, og nødvendige tiltak dersom den ikke er registrert, hvilke parametre man forventer å finne på stasjonen, og om det skal foretas noen form for konvertering av parametre under innlastingen.

Innlastingen mot HL skjer gjennom et overbygg til SQL*Loader hvor diverse logger føres og aksjoner tas utfra status i disse loggene. Under innlastingen føres statistikk over parameter-forkomster, og etter fullført innlasting ombyttes kolonnerekkefølgen i tabellen i henhold til statistikken for å oppnå maksimal plass-utnyttelse. En mer teknisk beskrivelse av innlastingen er å finne i [3].

Typisk eksperiment med innlasting (16069 rader) tok ca. 5.5 minutter. Arbeidsfordelingen ble da fordelt som følger i innlastingssystemet:

Kontroll:	1 min 04 sek (ca. 250 rader pr. sek)
Innlasting AL:	2 min 16 sek (ca. 120 rader pr. sek)
Innlasting HL:	2 min 16 sek (ca. 120 rader pr. sek)

Til tross for at kontrollen inneholder en rekke gjøremål består flaskehalsen i bruk av SQL*Loader. Eksperimentene benyttet seg av commit-point (bekreftelse på tabell-lagring) for hver 64'de innlastede rad. Det er mulig at yteevnen kan justeres noe ved å justere denne faktoren.

Under eksperimentering med innlasting oppdaget man at kolonnerekkefølgen ikke var likegyldig. Dersom kolonner uten fysisk innhold befant seg midt i tabellen viste det seg at databasen likvel avsatte plass. Hvis man på den annen side valgt å plassere disse kolonnene ytterst til høyre i tabellen ble dette unngått.

Et eksperiment med lagring av to identiske tabeller, en med vilkårlig parameter-rekkefølge (K00001) og en med optimalisert rekkefølge (K00002) ga følgende resultat:

NAVN	BYTES	BLOCKS
K00001	5771264	1409
K00002	3833856	936

En block = 4096 (2^{12}) bytes.

Prosentvis plass spart er:
(1409-936)/1409=0.3357 (33.57%).

6. VIDERE ARBEID

Året 1992 var preget av metodeutvikling og spesifikasjoner for databaseprosjektet. Mye av det teoretiske arbeidet for innlasting av data og programvareutvikling er dermed lagt. En rekke eksperimenter er gjort som gir indikasjoner på hvordan dette bør gjøres på en best mulig måte, og mye av utviklingsarbeidet er allerede godt i gang.

Første halvår i 1993 vil i utstrakt grad bli brukt til teoretisk og praktisk arbeid i forbindelse med innlasting av data, kontroll av data og viderebehandling av data. Sikkerhetsrutiner i forbindelse med databasen er dessuten under stadig utvikling.

6.1 Innlasting av data

Delprosjektet som behandler innlasting av data baserer seg sterkt på arbeider dokumentert i [3]. Dataflytdiagrammene benyttes som arbeidstegninger for programvareimplementasjon. En annen hovedoppgave er kartlegging og rørlegging av data fra utenverdenen inn til Klimaavdelingen. Belastning og kapasitet studeres.

En synlig konsekvens av det praktiske prosjektarbeidet vil være design og implementasjon av menyer og skjermbilder til bruk for punching, kontroll og retting av data.

Brukere ved Klimaavdelingen blir i mer utstrakt grad trukket inn i databasen etterhvert som databaseprosjektet behandler rutiner som angår den enkeltes arbeid.

6.2 Kontroll av data

Kontroll av data ivaretas av et eget prosjekt hvor man forsøker å kombinere de tradisjonelle kontrollteknikker benyttet ved Klimaavdelingen med metoder fra statistikk og numerisk analyse. Kontrollproblemet angripes delvis fra en abstrakt matematisk vinkel og delvis fra en konkret geofysisk vinkel.

Man er spesielt interessert i hvordan automatisk beslutningsstøtte kan integreres i kontrollsystemet, og hvordan på en mest mulig effektiv og sikker måte skape kommunikasjon mellom programenheter med ansvar for kontroll, retting og flagging av data.

I tillegg til automatiske hjelpemidler for kontroll til støtte for retting av data vil man sikte mot å ta i bruk grafiske hjelpemidler for diverse former for manuell kontroll.

6.3 Utlasting og bearbeiding av data

Systematisering og implementasjon av behandlingssystem for geofysisk data er tema for det siste av de tre store løpende prosjekter. Prosjektgruppen analyserer hvilke behandlingsmuligheter som er tilgjengelige pr. 1992 og benytter dette som basis for programutvikling i det nye systemet.

Sentralt i arbeidet finner man konstruksjon av utplukksalgoritmer, metoder for statistisk og geofysisk behandling, konstruksjon av programvare og integrasjon av kommersielle systemer.

Resultatene av prosjektet vil få synlige konsekvenser for forespørsler, forskning og daglige rutiner. Brukere ved Klimaavdelingen vil bli informert etterhvert som det praktiske arbeid skrider frem. Brukere vil bli trukket inn i beslutningsprosessene i situasjoner hvor konsekvensene av arbeidet vil være av sentral betydning.

REFERANSELISTE

[1] Notat 22/90 KLIMA av 09.04.90 "Kravspesifikasjon til ny database/maskin".

[2] DNMI-rapport 32/91 KLIMA "Database/Maskin-prosjektet i Klimaavd. Informasjonsmodell, flagging og kontroller. Status pr. 30.06.91"

[3] DNMI-rapport 40/92 KLIMA "Database-prosjektet i Klimaavd. Etablering av valgt datastruktur på Typhoon. Delprosjekt 3."

[4] DNMI-rapport 42/92 KLIMA "Database-prosjektet i Klimaavd. Utarbeiding og testing av ulike datastrukturer på Typhoon. Delprosjekt 2."

[5] DNMI-rapport 44/92 KLIMA "Databaseprosjektet i Klimaavdelingen. Standarder for systemutvikling. Delprosjekt 4."

[6] DNMI-rapport 45/92 KLIMA "Database-prosjektet i Klimaavdelingen. Kvalitetsstyring for prosjektarbeid. Delprosjekt 5."

[7] Overordnet prosjektplan av 11.11.91, versjon 1.0, "Prosjektplan for databaseprosjektet ved Klimaavdelingen"

[8] Overordnet prosjektplan av 20.11.92, versjon 6.0, "Prosjektplan for databaseprosjektet ved Klimaavdelingen"

[9] Kvalitetssikringshåndbok for Databaseprosjektet - Klimaavdelingen 09.12.92, rev. 1.0.