Towards an ISO-9000 standard for system development by DRIFT

September 7th, 1998

Petter Øgland
Towards an ISO-9000 standard for system development by DRIFT

Petter Øgland

DNMI - Climatology Division

The purpose of the computer program DRIFT is to give computer assistance to the development and maintenance cycle of the KLIBAS database system. The program contains algorithms for finding problems or latent problems with respect to program scheduling conflicting with Oracle availability, critical use of disk space on Dobserver, loading statistics and priority lists for programming and further development. The program generates output for the KLIBAS statistics report.

This is the first documented version of DRIFT although the program has been in use since April 1995. In order to reach an ISO-9000 level for the program development, the five level Software Engineering Institute (SEI) method is being used. The final level should be fairly close to ISO-9000.

Petter Øgland
Research Scientist

Bjørn Aune
Head of Division
1. PROBLEM FORMULATION AND SPECIFICATION
The purpose of the computer program DRIFT is to give computer assistance to the development and maintenance cycle of the KLIBAS database system. The program contains algorithms for finding problems or latent problems with respect to program scheduling conflicting with Oracle availability, critical use of disk space on Dserver, loading statistics and priority lists for programming and further development. The program generates output for the KLIBAS statistics report.

1.1 Problem formulation
The problem formulation is given by first a description of input, output and operation required by the program and then a list of the most frequently occurring problems caused by runs and test runs of the program so far.

INPUT: Year and month to be examined.
OUTPUT: Statistics for daily check and information for the monthly KLIBAS statistics report.
ALGORITHM: Reading program logs and other relevant information for the DRIFT system, making analysis and summarising the results.

1.2 Problem specification
In order to make sure that a solution can be achieved, the problem is analysed in order to find out whether it (1) does in fact have a solution, (2) the solution is unique and (3) whether the solution is stable, in other words if the solution is continuously dependable on the problem.

1.2.1 Existence of solution
The DRIFT program is run on a daily basis by KAPO in order to make the daily work of program maintenance and development systematic. The program consists of four major parts. The first part gives statistics for the KLIBAS project development, the second part contains information on Dserver and Oracle statistics, the third part contain statistics for the programs while the fourth and last part show plans for the next month.

The statistics generated by the program are read in consequence order, beginning with project statistics, showing productivity, quality in terms of solving problems relevant to the project and statistics showing how much time is spent on each particular task.

After verifying that project progress is under control, the system is analysed by lists showing the size of the KLIBAS system, defects in the system and how hard the system is working. The lists should be sufficient to decide which problems should be solved immediately. After the general lists, statistics and information concerning the different programs should checked to see if they appear to be working normally.
Finally a plan for the next month is presented.

1.2.2 Uniqueness of solution
Algorithms are made in order to make priority lists and quality statistics in such a manner that there should be no confusion about which problems are the most serious. Only problems that are absolutely necessary to solve are considered.

1.2.3 Stability of solution
The stability of the DRIFT program consists in how well the program reflects the best mental way to handle the development and maintenance tasks. Minor adjustments in the DRIFT program should only result in minor alterations with the KLIBAS statistics.

1.3 Runtime warnings
Below is a histogram presenting the distribution of problems. Only non-solved or non-ignored problems are accounted for.

![Histogram showing problem distribution](image)

<table>
<thead>
<tr>
<th>Problem</th>
<th>Description</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem 1</td>
<td>The DRIFT program should be remodelled after by REI or ISO-9000</td>
<td>61</td>
</tr>
<tr>
<td>Problem 2</td>
<td>DATAUT-RA Starjoeramme should be eliminated from WMS</td>
<td>57</td>
</tr>
<tr>
<td>Problem 3</td>
<td>The AMS section should be adusted by perhaps printing auto_backup.sys instead of list1</td>
<td>54</td>
</tr>
<tr>
<td>Problem 4</td>
<td>Plot wait must be implemented</td>
<td>52</td>
</tr>
<tr>
<td>Problem 5</td>
<td>The goal should be to produce a fully automated project report with PERT, GANTT, etc. a</td>
<td>48</td>
</tr>
<tr>
<td>Problem 6</td>
<td>The maintenance report should be systematically proof read</td>
<td>48</td>
</tr>
<tr>
<td>Problem 7</td>
<td>Program is running in test mode</td>
<td>37</td>
</tr>
</tbody>
</table>

The total count of non-solved or non-ignored problems adds up to 2254. The list show problems occuring the highest frequency. The plot below shows the day-by-day number of driftmatically reported errors and warning during the last twelve months.

![Time plot showing problems day by day](image)

A total of 2252 warnings have been recorded for the program DRIFT.
2. SYSTEM ANALYSIS AND DESIGN
Do we know any related problems which could simplify the design of the program?

The first documented attempts at making standard for system developments in the KLIBAS project are described in report no. 44/92 KLIMA. The report no. 45/92 KLIMA contained the groups plans for quality management of the project.

Some of the first methods in order to make work more systematic was by establishing a Quality Assurance Officer (KLIBAS-report nos. 33/94, 34/94 and 03/95) and a User Group (KLIBAS-report no. 27/95 and 07/97).

While there had been written status report for the KLIBAS project annually and sometimes biannually since 1991, a new series of monthly system statistics was introduced April 10th 1995 (KLIBAS-note no. 02/95) and have been published every month since.

The layout for these reports have been continuously revised. In May 1997, however, the first program that was generate with the sole purpose of updating the KLIBAS statistics report was published (KLIBAS-report no. 45/97). Other programs were shortly developed and documented (KLIBAS-report no. 46/97, 49/97, 50/97 and 53/97).

All the time, a program DRIFT had been responsible for producing and assembling statistics. In December 1997 the present version of this program was established.

2.1 System design at high level
The DRIFT program is executed on a daily basis. The program does some analysis but part of the analysis tasks are distributed among other programs such as ORACLE_SHUTDOWN (report no. xx/97 KLIBAS) and GRIM_REAPER (report no. xx/97 KLIBAS). The program ERRIDE, PRIPLAN and ERRFIX are used for updating and maintaining the priority list based on program errors and warnings.

In addition to the programs represented in the figure above DRIFT also makes system calls to the programs METARTID, SKRIVPLAN and PLOTLOG none of which are documented in the KLIBAS report series. The DRIFT program also assembles statistics from a number of programs, including SYNO_INN, AUTO_INN, SYNO_KONTR, PIO_INN, KLIMA_KONTR and many others central to the KLIBAS sys-
New ideas are constantly being added to the DRIFT working agenda and makes the DRIFT program often ranging in the upper half of the priority list.

**What is the current system doing?**
1. Project statistics are collected and presented. Project effort is plotted by use of a system call to PLOTLOG.
2. KLIBAS system statistics are collected and presented. Time diagram and frequency plot of errors from the err-files are made by system call to ERRIDE. The priority plan is made up by a system call to PRIPLAN. Daily log of programming is made by a system call to ERRFIX.
3. In order to collect and present the amount of space left on Disk, a system call to GRIM_REAPER is made.
4. Oracle statistics are based on a system call to ORACLE_SHUTDOWN.
5. For each section containing statistics for each category of weather stations, log files are being read for major programs related to these categories.

**What extra features should be included in the new system?**
1. There should be a display of the daily development of system size and system effort in addition to system quality.
2. The DRIFT program should be iteratively redesigned in order to make a SEI approach towards ISO-9000.
3. The DRIFT program should produce a fully automated project report with PERT, GANTT, etc. automatically estimated and updated.
4. There should be a better control of the VNN routine. The routine apparently broke down August 29th 1998 as indicated in letter from Per-Tore Soervoll.
5. The principle of system/data monitoring by DRIFT, SYNOP and SYNOP_KONTR should be explained in the report.
6. The principle of system/data monitoring by DRIFT, AUTO and MKK/ADK should be explained in the report.

**What constraints (such as response time) must the new system satisfy?**
Presently there are no constraints to the system.

### 2.1.1 Main function structure
The program consists of a call to a function collecting system statistics and another call a function analyzing and printing the results in a suitable format.
The graphs below shows the number of executions per month and execution time for the program as recorded so far.

2.2 Analysis and design at file level
The system drift is stored on directory /usr2/klima/people/kapo/drift and consists for the moment of the following files:

```
total 10584
drwxr-xr-x 5 kapo klima 45 Apr 14 11:55 AANDERAA
drwxr-xr-x 3 kapo klima 21 Apr 14 11:55 ADM
drwxr-xr-x 4 kapo klima 33 Apr 14 11:55 APPEND
drwxr-xr-x 7 kapo klima 69 Apr 14 11:55 AUTO
drwxr-xr-x 3 kapo klima 21 Apr 14 11:55 DATAUT
drwxr-xr-x 6 kapo klima 57 Apr 14 11:56 DRIFT
drwxr-xr-x 6 kapo klima 57 Apr 14 11:56 FERM
drwxr-xr-x 5 kapo klima 45 Apr 14 11:56 INFO
drwxr-xr-x 4 kapo klima 33 Apr 14 11:56 INNLIEDN
drwxr-xr-x 5 kapo klima 70 Apr 14 11:56 KLIMA
drwxr-xr-x 3 kapo klima 21 Apr 14 11:56 LINKE
drwxr-xr-x 7 kapo klima 69 Apr 14 11:56 MDS
```
draXx-2r-x  6  kapo  klima  57  Apr  14  11:56  METAR
draXx-2r-x  7  kapo  klima  69  Apr  14  11:56  PLUMATIC
draXx-2r-x  7  kapo  klima  69  Apr  14  11:56  PRECIP
-draXx-2r-x  1  kapo  klima  354  Sep  2  10:36  auto.log
-draXx-2r-x  1  kapo  klima  4096  May  7  09:27  bin
-draXx-2r-x  1  kapo  klima  84184  Jul  21  19:12  cont1.txt
-draXx-2r-x  1  kapo  klima  47890  Jul  21  18:08  cont2.txt
-draXx-2r-x  1  kapo  klima  192425  Jul  21  19:10  cont3.txt
-draXx-2r-x  2  kapo  klima  12288  Sep  7  14:14  dat
-draXx-2r-x  1  kapo  klima  1239116  Sep  7  14:14  drift
-draXx-2r-x  1  kapo  klima  136943  Sep  2  10:21  drift.C.old
-draXx-2r-x  1  kapo  klima  164605  Sep  7  14:14  drift:c
-draXx-2r-x  1  kapo  klima  3069  Sep  7  13:48  drift.csh
-draXx-2r-x  1  kapo  klima  5960  Sep  7  13:49  drift.err-imp
-draXx-2r-x  1  kapo  klima  24337  Sep  7  14:16  drift.err-01
-draXx-2r-x  1  kapo  klima  16800  Sep  7  14:16  drift.err-02
-draXx-2r-x  1  kapo  klima  104108  Sep  7  14:16  drift.err-03
-draXx-2r-x  1  kapo  klima  64664  Sep  7  14:16  drift.err-04
-draXx-2r-x  1  kapo  klima  43940  Sep  7  14:16  drift.err-05
-draXx-2r-x  1  kapo  klima  82854  Sep  7  14:16  drift.err-06
-draXx-2r-x  1  kapo  klima  56799  Sep  7  14:16  drift.err-07
-draXx-2r-x  1  kapo  klima  223182  Sep  7  14:16  drift.err-09
-draXx-2r-x  1  kapo  klima  0  Sep  7  14:16  drift.err-10
-draXx-2r-x  1  kapo  klima  0  Sep  7  14:16  drift.err-11
-draXx-2r-x  1  kapo  klima  67  Sep  7  14:16  drift.err-12
-draXx-2r-x  1  kapo  klima  110  Apr  22  14:23  drift.fix.01
-draXx-2r-x  1  kapo  klima  0  Apr  21  14:23  drift.fix.02
-draXx-2r-x  1  kapo  klima  181  Apr  21  14:23  drift.fix.03
-draXx-2r-x  1  kapo  klima  71  Sep  7  14:04  drift.fix.04
-draXx-2r-x  1  kapo  klima  442  Apr  30  09:02  drift.fix.05
-draXx-2r-x  1  kapo  klima  0  Sep  7  14:04  drift.fix.06
-draXx-2r-x  1  kapo  klima  582  May  29  11:10  drift.fix.07
-draXx-2r-x  1  kapo  klima  85  Sep  7  14:04  drift-fix.06
-draXx-2r-x  1  kapo  klima  413  Jun  25  13:43  drift-fix.06
-draXx-2r-x  1  kapo  klima  321  Sep  7  14:04  drift-fix.07
-draXx-2r-x  1  kapo  klima  683  Jul  30  18:13  drift-fix.07
-draXx-2r-x  1  kapo  klima  71  Sep  7  14:04  drift-fix.08
-draXx-2r-x  1  kapo  klima  443  Sep  7  14:04  drift-fix.09
-draXx-2r-x  1  kapo  klima  132  Apr  21  14:23  drift-fix.12
-draXx-2r-x  1  kapo  klima  20987  Sep  7  14:16  drift-log
-draXx-2r-x  1  kapo  klima  0  Sep  2  10:43  drift-log.01
-draXx-2r-x  1  kapo  klima  0  Sep  2  10:43  drift-log.02
-draXx-2r-x  1  kapo  klima  0  Sep  2  10:43  drift-log.03
-draXx-2r-x  1  kapo  klima  0  Sep  2  10:43  drift-log.04
-draXx-2r-x  1  kapo  klima  0  Sep  2  10:43  drift-log.05
-draXx-2r-x  1  kapo  klima  0  Sep  2  10:43  drift-log.06
-draXx-2r-x  1  kapo  klima  0  Sep  2  10:43  drift-log.07
-draXx-2r-x  1  kapo  klima  0  Sep  2  10:43  drift-log.08
-draXx-2r-x  1  kapo  klima  7  Sep  7  14:16  drift-log.09
-draXx-2r-x  1  kapo  klima  7  Sep  7  14:16  drift-log.10
-draXx-2r-x  1  kapo  klima  0  Sep  2  10:43  drift-log.11
-draXx-2r-x  1  kapo  klima  0  Sep  2  10:43  drift-log.12
-draXx-2r-x  1  kapo  klima  253424  Sep  7  14:14  drift.o
-draXx-2r-x  1  kapo  klima  72864  Sep  7  14:16  drift.out
-draXx-2r-x  1  kapo  klima  149269  Sep  7  14:10  drift.pc
-draXx-2r-x  1  kapo  klima  149269  Sep  7  14:14  drift.pc.old
-draXx-2r-x  1  kapo  klima  474  Sep  7  09:46  drift.pln
-draXx-2r-x  1  kapo  klima  4867  Sep  7  11:26  drift.sql
Whenever the program drift.csh is executed, it does a comparison between the files drift.pc and drift.pc.old. If the files differ, drift.pc is copied to drift.pc.old and compiled using the make facility and auxhiliary file proc.mk for getting the Oracle interface right.

While executing, the files symptoms, drift.err.01, drift.err.02, ..., drift.err.12 are updated. General output from drift is written to drift.out. All information directed to stderr is redirected to the file symptoms. At the end of execution of drift.csh, the symptoms file is investigated, and if there seems to be a reason to do so, an e-mail message is sent to kapo.

The files drift.log, drift.err.01, drift.err.02, ..., drift.err.12 are updated at the end of session. The file drift.log contains executional statistics, while the other files contain summary of errors and warnings recorded for the month of execution.

The source code drift.pc is made up of 4766 lines, distributed over 122 functions. Complete source code listing is given in the appendix.
2.3 System design at low level
Below is a description of the main functions used by the program.

2.3.1 Presenting project statistics
Using the work breakdown structure (WBS) as basis for generating project statistics, algorithms for measuring productivity, quality and effort are constructed and presented by curves.

```c
int adm(FILE *fq,int chapter)
{
    FILE *fp;
    int prev_month,prev_year;
    prev_year=ear0;
    prev_month=mmd0-1;
    if(!prev_month)
        prev_month=12;
    prev_year--;
}

fprint(fq,"\fB\%d. KLIBAS PROJECT DEVELOPMENT\fR\%d.chapter");
fprint(fq,"\fB sp 00\fB\%d. The KLIBAS project consists of developing and maintaining the parts of the KLIBAS database system that is defined and planned in the KLIBAS statistics notes. The statistics below are based on the work breakdown structure (WBS) given in appendix one. The three metrics Productivity, Quality and Efficiency for software projects are slightly adapted from Yourdon [1] with emphasis on the process as described by Nitra [2].0\fR\%d.\n",sp,0);
fprint(fq,"\fB sp 00fB\%d. A measure for project productivity\fR\%d.chapter");
fprint(fq,"\fB sp 00h curve below shows the number of finished tasks on the WBS. Measures are taken day by day since early May 1998 and by the end of the month prior to that.0\fR\%d.\n",sp,0);
if(plotOppgaver(fq,'Specific'))goto error;

fprint(fq,"\fB sp 00fB\%d. A measure for project quality\fR\%d.chapter");
fprint(fq","sp 00he curve below is based on the formula (T-F)/T where T is the total number of defined tasks according to WBS at the sample point while F is the number of finished tasks. The formula hence gives a measure for the relative number of unfinised tasks day by day which is also a sort of defect measure or quality measure.0\fR\%d.\n",sp,0);
if(plotOppgaver(fq,'Relative'))goto error;

fprint(fq,"\fB sp 00fB\%d. A measure for project efficiency\fR\%d.chapter");
fprint(fq,"sp 00elow is a plot showing how many days the average work task has been on the WBS. Dashed lines represent one standard deviation.0\fR\%d.\n",sp,0);
sprintf(buf,"\fB sp 00d\%d. /usr2/klima/people/kapo/drift2/binsrc/plotlog %0d %0d /usr2/klima/people/kapo/drift2/FREM/log/oppgaver.log 3 > %s/%s.tmp",ear0,mmd0,path.name);
if(system(buf))
    fprint(stderr,"ERROR: Cannot execute %s0,buf); goto error;
}
sprintf(buf,"%s/%s.tmp",path.name);
fp=fopen(buf,"r");
if(NULL==fp)
    fprint(stderr,"ERROR: Cannot open %s0,buf); goto error;
```
2.3.2 General KLIBAS system statistics
Several system statistics such as system size, system quality and system effort is collected and presented.

```c
int klibas(FILE *fp,int chapter){
    int i;
    FILE *fp;
    char text[100]={
        "sp 00ps 80vs 10", //PS 6.2,
        "movewd = 1", "moveht = 1", "boxwid = 20", "boxht = 4", "[", "boxwd = 5", "boxht = 1", "movewd = 1", "moveht = 1", "lineht = 1", "COLLECT": box "move from COLLECT.right right", "QUALITY": box "move from QUALITY.right right", "BACKUP": box "move from QUALITY.bot down",
    }
    ...
struct progs{
    int data_inn, kontroll, data_ut, backup, adm, info;
    int others;
    int total;
} progs;
progs.data_inn=0;
progs.data_ut=0;
progs.kontroll=0;
progs.backup=0;
progs.info=0;
progs.adm=0;
progs.others=0;
progs.total=0;
if(collect_err_files()) goto error;
sprintf(buf, "%s/\%s.err-inp", path, name);
fp=fopen(buf, "r");
if(NULL==fp){
    fprintf(stderr, "ERROR: Cannot open %s.buf
    goto error;
}
while(NULL!=fgets(buf, BUPSIZ, fp)){
    if(NULL!=strstr(buf, "_ut")) progs.data_ut++;
    else if(NULL!=strstr(buf, "vsuke")) progs.data_ut++;
    else if(NULL!=strstr(buf, "ut1")) progs.data_ut++;    
else if(NULL!=strstr(buf, "rdut")) progs.data_ut++;
else if(NULL!=strstr(buf, "inn")) progs.data_inn++;    
else if(NULL!=strstr(buf, "vnn")) progs.data_inn++;    
else if(NULL!=strstr(buf, "al2hia")) progs.data_inn++;    
else if(NULL!=strstr(buf, "vind_reg")) progs.data_inn++;    
else
else if(NULL!=strstr(buf,"mkk"))progs.kontroll++; 
else if(NULL!=strstr(buf,"s-t-f"))progs.kontroll++; 
else if(NULL!=strstr(buf,"kontr"))progs.kontroll++; 
else if(NULL!=strstr(buf,"cont")progs.kontroll++; 
else if(NULL!=strstr(buf,"met/"))progs.kontroll++; 
else if(NULL!=strstr(buf,"backup")progs.backup++; 
else if(NULL!=strstr(buf,"export")progs.backup++; 
else if(NULL!=strstr(buf,"mutable")progs.info++; 
else if(NULL!=strstr(buf,"info")progs.info++; 
else if(NULL!=strstr(buf,"adm")progs.adm++; 
else if(NULL!=strstr(buf,"drift"))progs.adm++; 
progs.total++; 
}
progs.others=progs.total-{progs.data_inn+progs.data_ut+progs.kontroll+progs.adm+progs.backup};
fclose(fp);
fprintf(fp,".bp0f&d. KLIBAS - CLIMATOLOGICAL DATABASE SYSTEM\r0,chapter);
fprintf(fp,".sp 00);
fprintf(fp,"At the end of %s %d the author of this note has been responsible for %d computer programs running on the KLIBAS database system.0.month[md],aar,progs.total); 
fprintf(fp,"%d programs are used for loading and registering observations, %d programs are used for quality control, %d programs are used for backup and export, %d programs are used for updating and changing info archive and database structures, %d programs are used for end user products and %d programs are used for administrative and auxiliary purposes.0.progs.data_inn,progs.kontroll,progs.backup,progs.info,progs.data_ut,progs.adm+progs.others); 
fprintf(fp,".sp0); 
for(i=0;NULL=text[i];i++) 
fprintf(fp,"%s0,text[i]);
fprintf(fp,"The system indicated above could be further expanded by adding external producer of weather observations and external consumer of weather- or climate statistics.0sp0); 
fprintf(fp,".sp 00fB&d.1. Problems still not solved %s %d.\r0,chapter,month[md],aar0);
fprintf(fp,".sp 00he curves below show the total number of errors/warnings recorded day by day for the last 31 days. The total numbers of errors with respect to each program during this period is displayed by histograms.0);
sprintf(buf,"/usr2/klima/people/kapo/drift2/erride/erride.cash %04d %02d 31",aar0,md0);
if(system(buf)){
fprintf(stderr,"ERROR: Cannot execute %s0.buf); 
goto error;
}
sprintf(buf,"/usr2/klima/people/kapo/drift2/erride/erride.cash %04d",md0); 
fp=fopen(buf,"r"); 
if(NULL==fp){
fprintf(stderr,"ERROR: Cannot open %s0,buf); 
goto error;
}
while(NULL!=fgets(buf,BUFSIZE,fp)) {
fprintf(fp,"%s",buf);
}
fclose(fp);
fprintf(fp,"sp0);
fprintf(fp,"The problems are ordered by importance and sequence of execution.0);
printf(buf,"/usr2/klima/people/kapo/drift2/priplan/priplan.cash %04d %02d
"};
# 2.3.3 General klibas system statistic

Several system statistics such as system size, system quality and system effort is collected and presented.

```c
int systemStat(FILE *fq, int chapter){
    FILE *fp;
    int i;
    char *text[100]={
        "The figure below is intended give a general overview of how different categories of weather observations flow through the KLIBAS system and are stored in the main storage unit HL.",
        "SP",  
        "SP 00ps 80vs 10", 
        "PS 6.3", 
        "boxwid = 5", 
        "boxht = 1", 
        "movewid = 1";
```
fp=fopen(buf,"r");
if(NULL==fp){
    fprintf(stderr,"ERROR: Cannot open %s0,buf");
goto error;
}
while(NULL!=fgets(buf,BUFSIZE,fp)){
    fprintf(fq,"%s",buf);
}
fclose(fp);
for(i=0;NULL!=text[i];i++)
    fprintf(fq,"%s0,text[i]";;
fprintf(fq,".bp0fs#d.1 Disk usage monitoring for /usr/people\fR0sp 00,chapter
sprint(buf, /usr/people/kapo/grim_reaper/grim_reaper.csh %0d %02d > /dev/null 2>
/dev/null",ear0,md00);
if(system(buf))
    fprintf(stderr,"ERROR: Cannot execute %s0,buf");
goto error;
}
sprintf(buf,"/usr/people/kapo/grim_reaper/grim_reaper.txt")
fp=fopen(buf,"r");
if(NULL==fp){
    fprintf(stderr,"ERROR: Cannot open %s0,buf");
goto error;
}
while(NULL!=fgets(buf,BUFSIZE,fp)){
    fprintf(fq,"%s",buf);
}
fclose(fp);
return 0;
error:
    fprintf(stderr,"ERROR in systemStat0");
    return 1;
}

2.3.4 System statistics for the SYNOP routine
Statistics for vital programs and system elements are collected and presented.

int archiveSynop(FILE *fq,int chapter){
    int i;
    char *text[100]={
        "..sp",
        "$00ps 80ve 10",
        "$PS 6.2",
        "$boxwid = 5",
        "$boxht = 0.67",
        "$movewid = 1",
        "$moveht = 0.5",
        "$lineht = 0.5",
        "$EDB: box",
        "$move from EDB.right right",
        "$ALA: box",
        "$move from ALA.top up",
        "$SYNOP: box",
        "$move from ALA.bot down",
        "$PIO: box",
        "$move from PIO.bot down",
        "$MRPAR: box",
        "$move from ALA.right right",
    };
...
"arrow from EDB.right to SYNOP.left",
"arrow from EDB.right to ALA.left",
"arrow from EDB.right to PIO.left",
"arrow from EDB.right to METAR.left",
"move from ALA.right right",
"TELE: box",
"arrow from SYNOP.right to TELE.left",
"arrow from ALA.right to TELE.left",
"arrow dotted from PIO.right to TELE.left",
"arrow dotted from METAR.right to TELE.left",
"move from TELE.right right",
"KLIMA: box",
"move from KLIMA.top up",
"VA: box",
"arrow from TELE.right to VA.left",
"arrow from TELE.right to KLIMA.left",
"",
",.sp 00ps 110vs 13",
",.sp",
NULL
});
FILE *fp;
struct elem{
  char *name;
  char *explain;
}
elem[12]={(]("temperature",
  "TN", "max temperature",
  "TX", "min temperature",
  "PO", "air pressure at station level",
  "P", "air pressure at sea level",
  "N", "clouds",
  "RR", "precipitation",
 NULL,NULL
});
fprintf(fp,".bp0f8%d. SYNOPTIC MANUAL WEATHER STATIONS (SYNOP/TELE) \fr0sp 00,chapter);
fprintf(fp,.sp 00he most critical factors in the SYNOP/TELE routine, investigated
here, are the way observations are entering the SYNOP/TELE tables, the way automatic
interpolation and correction of observations is carried out and how the monthly
KA_H_STAT statistics are coming along.0);
for(i=0;NULL!=text[i];i++)
  fprintf(fp,"%s0.text[i]0;
fprintf(fp,"Dotted lines in the figure above indicates that dataflow is not yet es-
ablished.0);
fprintf(fp,".sp0f8%d.1 Monitoring the SYNO_INN system\fr0,chapter);
fprintf(fp,".sp 00);
ssprintf(buf,\"/usr/people/kapo/synop/syno_kontr/syno_test/syno_test.txt.802d\",mnd0);
fp=fopen(buf,"r");
if(NULL==fp){
  fprintf(stderr,"ERROR: Cannot open \s0, buf);
goto error;
}
while(NULL!=fgets(buf,BUFSIZE,fp)){
  fprintf(fp,"%s",buf);
}
fclose(fp);
fprintf(fp,".sp0f8%d.2 Monitoring the SYNKO_KONTR system\fr0,chapter);
fprintf(fp,".sp 00he SYNKO_KONTR system consists of several programs for used for au-
tomatical and manual quality control and treatment of observations stored in the data-
table TELE. @sp0); 
sprintf(buf,"/usr/people/kapo/synop/syno_kontr/syno_kontr.txt"); 
fp=fopen(buf,"r"); 
if(NULL==fp){ 
 fprintf(stderr,"ERROR: Cannot open %s0.buf"); 
 goto error; 
} 
while(NULL!=fgets(buf,BUFSIZ,fp)){ 
 fprintf(fp,"%s",buf); 
} 
fclose(fp); 
fprintf(fp,".sp0elow is a performance analysis of the INTERPOL2 automatic interpola-
tion routine is given by a list of the ten greatest RMSEs (root mean square errors) ob-
served for each meteorological element during %s0.month\{\nd0\}); 
 for(i=0;NULL!=elem[i].name;i++){ 
 fprintf(fp,".sp0fBInterpolation RMSE for %s (%s)\{\fr0,elem[i].explain,el-
 em[i].name\}); 
 sprintf(buf,"/usr/people/kapo/synop/syno_kontr/interpol2/interpol2.%s.\%02d\{,el-
 em[i].name,\nd0\}); 
fp=fopen(buf,"r"); 
if(NULL==fp){ 
 fprintf(stderr,"ERROR: Cannot open %s0.buf"); 
 goto error; 
} 
while(NULL!=fgets(buf,BUFSIZ,fp)){ 
 fprintf(fp,".sp 018s",buf); 
} 
fclose(fp); 
} 
fprintf(fp,".sp0elow are statistics from the program INTERPOL_P0 (report no. 46/98 
KLIBAS) that is constructed in order to improve the air pressure interpolation done by 
INTERPOL2.0); 
sprintf(buf,"/usr/people/kapo/synop/syno_kontr/interpol_p0/interpol_p0.txt"); 
fp=fopen(buf,"r"); 
if(NULL==fp){ 
 fprintf(stderr,"ERROR: Cannot open %s0.buf"); 
 goto error; 
} 
while(NULL!=fgets(buf,BUFSIZ,fp)){ 
 fprintf(fp,"%s",buf); 
} 
fclose(fp); 
fprintf(fp,".sp0fB%d.3 Monitoring the KA_H_STAT routine\{\fr0,chapter\); 
fprintf(fp,".sp 00one of the main purposes of the SYNOP/TELE routine is to generate 
monthly climate statistics as soon as observations are stored on the KLIBAS database 
system. Numerical values for this monthly product is generated by the program 
KA_H_STAT. A program CHECK_H_STAT is run as a part of the SYNOCONTR system in order 
to monitor problems arising due to data quality used for the KA_H_STAT program.0sp0); 
sprintf(buf,"/usr/people/kapo/synop/syno_kontr/check_h_stat/check_h_stat.txt"); 
fp=fopen(buf,"r"); 
if(NULL==fp){ 
 fprintf(stderr,"ERROR: Cannot open %s0.buf"); 
 goto error; 
} 
i=0; 
while(NULL!=fgets(buf,BUFSIZ,fp)){ 
 fprintf(fp,".sp 00fB%d.\fr %s\{,\si\},buf); 
} 
fclose(fp);
2.3.5 System statistics for the METAR routine
Statistics for vital programs and system elements are collected and presented.

```c
int archiveMetar(FILE *fq, int chapter){
  FILE *fp;
  fprintf(fq, "%.sp0FB% d. 4 Monitoring the STATUT routine\fr0, chapter);
  fprintf(fq, "%.sp 00 other main purposes of the SYNOP/TELE routine is to generate daily statistics based on the last 30 days of weather observations in TELE. The statistics are produced by a program STATUT. A program CHECK_STATUT is used for monitoring problems arising due to data quality used for the STATUT program. 0sp00); 
  fprintf(buf, "/usr/people/kapo/synop/sync_kontr/check_statut/check_statut.txt*); 
  fp=fopen(buf, "r"); 
  if(NULL!=fp){
    fprintf(stderr, "ERROR: Cannot open %s0, buf); 
    goto error;
  }
  i=0;
  while(NULL!=fgets(buf, BUFSIZ, fp)){
    fprintf(fq, "%s", buf);
  }
  fclose(fp);
  fprintf(fq, "%.sp0F% d. 5 Monthly summary of problems in the SYNOP/TELE routine\fr0, chapter);
  fprintf(fq, "%.sp 00 on the beginning of every month problems during production of the monthly climatology statistics are usually noted. Systematic digitalisation of such problems commenced July 9th 1998. 0sp00); 
  fprintf(buf, "/usr/people/kapo/synop/synop.inp*); 
  fp=fopen(buf, "r"); 
  if(NULL!=fp){
    fprintf(stderr, "ERROR: Cannot open %s0, buf); 
    goto error;
  }
  i=0;
  while(NULL!=fgets(buf, BUFSIZ, fp)){
    fprintf(fq, "%s", buf);
  }
  fclose(fp);
  return 0;
error:
  fprintf(stderr, "ERROR in archiveSynop00); 
  return 1;
}
```
if(system(buf)){
    fprintf(stderr,"ERROR: Cannot execute $s0,buf);
    goto error;
}

if(NULL==fp){
    fprintf(stderr,"ERROR: Cannot open $s0,buf);
    goto error;
}

while(NULL!=fgets(buf,BUFSIZE,fp)){
    fprintf(fp,"$s",buf);
}

fclose(fp);

if(system(buf)){
    fprintf(stderr,"ERROR: Cannot execute $s0,buf);
    goto error;
}

if(NULL!=fgets(buf,BUFSIZE,fp)){
    fprintf(fp,"$s",buf);
}

fclose(fp);

if(NULL!=fgets(buf,BUFSIZE,fp)){
    fprintf(fp,"$s",buf);
}

error:
    fprintf(stderr,"ERROR in archiveKlima\0");
    return 1;
}

2.3.6 System statistics for the KLIMA routine
Statistics for vital programs and system elements are collected and presented.

int archiveKlima(FILE *fp,int chapter){

    FILE *fp;
    fprintf(fp,"bp0fb\d. RESULTS FROM THE KLIMA ROUTINE\fr0sp 00,chapter\):
    fprintf(fp,"What is written below is a status of the routine according to program
KLIMA_R0NTR that makes daily checks and updates\0");
2.3.7 System statistics for the SAWS routine
Statistics for vital programs and system elements are collected and presented.

```c
int archiveSAWS(FILE *fq, int chapter)
{
    FILE *fp;
    fprintf(fq, "0bsp0fB%"d SEMI-AUTOMATED WEATHER STATIONS (SAWS)\"fR0sp 00,chapter);
    fprintf(fq, "0sp0fB%"d FIIO stations (\n    fprintf(fq, "0sp0fB%"d 00);    fprintf(buf,"/usr/people/kapo/klima/pio_inn/pio_inn.txt.%02d",mnd0);
    fp=fopen(buf, "r");    if(NULL==fp){
        fprintf(stderr, "ERROR: Cannot open %s0,buf);        goto error;
    }
    while(NULL!=fgets(buf, BUFSIZE,fp)){
        fprintf(fq,"%s",buf);
    }
    fclose(fp);
    return 0;
    error:
        fprintf(stderr,"ERROR in archiveKlima0);        return 1;
}
```

2.3.8 System statistics for the AWS routine
Statistics for vital programs and system elements are collected and presented.

```c
int archiveAuto(FILE *fq, int chapter){
    FILE *fp;
    fprintf(fq,".lp0fB$ed. AUTOMATIC WEATHER STATIONS (AWS) \fr0sp 00, chapter);  
    fprintf(fq,".sp0fB$ed.1 Realtime automatic weather stations\fr0sp 00, chapter);  
    sprintf(buf,\"/usr/people/kapo/auto/auto_inn/auto_inn.txt.%02d\",mnd0);  
    fp=fopen(buf,\"r\");  
    if(NULL==fp){
        fprintf(stderr,"ERROR: Cannot open %s0, buf);  
        goto error;  
    }
    while(NULL!=fgets(buf,BUFSIZE,fp)){
        fprintf(fq,"%s",buf);
    }
    fclose(fp);
    fprintf(fq,".sp0sily export is done by a cron job that places data files on the export disk area /usr/people/kapo/auto/auto_backup/export and comprise these.0);  
    sprintf(buf,\"/usr/people/kapo/auto/auto_backup/auto_backup.txt.%02d\",mnd0);  
    fp=fopen(buf,\"r\");  
    if(NULL==fp){
        fprintf(stderr,"ERROR: Cannot open %s0, buf);  
        goto error;  
    }
    while(NULL!=fgets(buf,BUFSIZE,fp)){
        fprintf(fq,"%s",buf);
    }
    fclose(fp);
    return 0;  
error:
    fprintf(stderr,"ERROR in archiveAuto());  
    return 1;  
}
```

### 2.3.9 Making plans for next month

The program tries to guess how much work that is likely to be done next month and lists the most vital problems to be solved.

```c
int newPlans(FILE *fq, int chapter){
    FILE *fp;
    int yyy,mmm,i;
    double count[13],tasks;
    yyy=ear0;
    mmm=mnd0+1;
    if(mmm>12){
        mmm=1;
        yyy++;
    }
    sprintf(buf,"%s .month[mmm];
    for(i=0;i<strlen(buf);i++)
        buf[i]=toupper(buf[i]);
    fprintf(fq,".lp0fB$ed. PLANS FOR %s \fr0sp 00, chapter,buff,yyy);  
    fprintf(fq,".sp 00statistics in chapter two shows the progression of problem solving by programming. As is illustrated in appendix 1, the solving of greater KLIRAS problems are often accompained with system documentation in the KLIRAS report series. By taking the average of the number of reports published during the last three months,
the number of expected greater problems to be solved in this manner for %s %d is estimated 0.month[mmm].yyy);
    fprintf(fp,".sp0\0e problem decided to work upon are normally selected from the list of unsolved problems in chapter two. A complete list of KLIBAS reports published so far is listed in appendix 2.0);
    fprintf(fp,".ts0 1.0); 
    sprintf(buf,"%s/dat/rapporter.txt",path);
    fp=fopen(buf,"r");
    if(NULL==fp){
        fprintf(stderr,"ERROR: cannot open %s0,buf);
        goto error;
    }
    mmm=mm0;
    yyy=aaa0-1;
    for(i=0;i<13;i++){
        count[i]=0.0;
        rewind(fp);
        sprintf(buf2,"%02d.%04d",mmm,yyy);
        while(NULL!=fgets(buf,BUFSIZE,fp)){
            if(NULL!=strstr(buf,buf2)\000NULL!=strstr(buf,"KLIBAS")\000NULL!=strstr(buf,"Report")\000)
                count[i]++;
            }
            if(i){
                if(count[i]<=1)
                    fprintf(fp,"%s %d%0f KLIBAS program documentation report 0.month[mmm].yyy,count[i]);
                else
                    fprintf(fp,"%s %d%0f KLIBAS program documentation reports 0.month[mmm].yyy,count[i]);
            }
            mmm++;
            if(mmm>12){
                mmm=1;
                yyy++;
            }
        }
    fclose(fp);
    fprintf(fp,".ts0");
    for(i=2;i<13;i++)
        count[i]=count[i];
    tasks=count[i]/12.0;
    fprintf(fp,".sp0\0e number of expected program documentation reports to be written during %s %d is %f.0.month[mmm].yyy,tasks); 
    fprintf(fp,".sp0\0e is a list of the %0f most important programming challenges for %s %d according to list in chapter two.0sp0,tasks.month[mmm].yyy);
    sprintf(buf,"%s2/priplain/priplain.%02d",path,mmm0);
    fp=fopen(buf,"r");
    if(NULL==fp){
        fprintf(stderr,"ERROR: cannot open %s0,buf);
        goto error;
    }
    i=0;
    while(NULL!=fgets(buf,BUFSIZE,fp)){
        if(NULL!=strstr(buf,"\fb")\002){
            if(count[i]<=((int)tasks)+2)
                fprintf(fp,".sp 023s",buf);
        }
    }

}
fclose(fp);
return 0;

error:
fprintf(stderr,"ERROR in newPlans0");
return 1;
}
3. SOFTWARE METRICS AND PROJECT HISTORY
As a measure for at what pace the project has been developing, the number of lines of program code on each run during September 1998 has been recorded and drawn on the curve below.

The maintenance log is updated manually on a daily basis. The complete log for all programs is displayed in monthly maintenance and performance statistics report.

Mon Sep 07 1998
A system specification for the present version of DRIFT was written. Minor problems and errors were corrected in the program.

Sat Sep 05 1998
The chapter on project statistics was redesigned and partly reprogrammed.

Wed Sep 02 1998
The KLIBAS statistics report was restructured. Chapter 1 was totally rewritten.

Tue Sep 01 1998
A first draft specification was written for DRIFT, based on the AUTO program specification.

Sat Aug 29 1998
Statistics from INTERPOL_P0 was added to the program.

Wed Jul 15 1998
The curve showing progression in problem solving is displayed by units of one day or measurement.

Fri Jul 10 1998
More diagrams were added to sections on the SYNOP routine explaining the topological data flow structure of the routine.

Thu Jul 09 1998
Diagrams were added to sections on the SYNOP routine.

**Wed Jun 03 1998**
Some lines on the general maintenance and development was included.

**Tue Apr 21 1998**
The program drift was reimplemented as a C++ program.

**Mon Mar 09 1998**
The program was edited and recompiled. Documentation was made up to date.

**Mon Mar 02 1998**
A mixup of dates for chapter four on programming priorities was mended.

**Fri Jan 02 1998**
The computer programme DRIFT2 was edited in order to produce correct data for December 1997.

**Thu Dec 04 1997**
The script drift2.csh was constructed in order to have better quality control over Maintenance/Performance report.

**REFERENCES**
Rapport 32/91 KLIMA "Database/maskin prosjektet i Klimaameldingen 1990-1991. Informasjonsmodell, flagging og kontroller. Status pr 30.06.91"
Rapport 40/92 KLIMA "Etablering av valgt datastruktur på Typhoon. Delprosjekt 3"
Rapport 42/92 KLIMA "Utarbeiding og testing av ulike datastrukturer på Typhoon. Delprosjekt 2"
Rapport 44/92 KLIMA "Standarder for systemutvikling. Delprosjekt 4"
Rapport 45/92 KLIMA "Kvalitetsstyring for prosjektarbeid. Delprosjekt 5"
Rapport 08/93 KLIMA "Sikkerhetsrutiner. Delprosjekt 7.8"
Rapport 03/94 KLIBAS "Databaseprosjektet i Klimaameldingen. Status pr 31.12.1993"
Rapport 10/94 KLIBAS "Databaseprosjektet i Klimaameldingen. KLIBAS systemoversikt-applikasjoner. Teknisk løsning, systemoversikt, meny, aksesstrøm, brukerdialog, applikasjonsarkitektur."
Rapport 17/94 KLIBAS "Rapport fra brukergruppen: Forslag til spesifisjon for data-ut-programmer"
Rapport 24/94 KLIBAS "Databaseprosjektet i Klimaameldingen. Status pr første halvår 1994"
Rapport 27/94 KLIBAS "Skisse til et generelt data-kontroll-system for geofysiske data"
Rapport 28/94 KLIBAS "Omlagging av databasenutining ved overgang fra Oracle6 til Oracle7"
Rapport 33/94 KLIBAS "Praktisk rutine for kvalitetssikring av programvare"
Rapport 34/94 KLIBAS "Kvalitetsbåndbok for databaseprosjektet i Klimaameldingen"
Rapport 03/95 KLIBAS "Kvalitetshåndbok for databaseprosjektet i Klimaameldingen. Del II"
Rapport 06/95 KLIBAS "Databaseprosjektet i Klimaameldingen. Status pr årsskifte 1994/95"
Rapport 22/95 KLIMA "KLIBAS - The DNMI Climatological Database System"
Rapport 27/95 KLIBAS "Brukergruppens arbeid 1994-95"
Rapport 01/96 KLIBAS "Databasegruppen 1995"
Rapport 04/96 KLIBAS "Automatisering av kvalitetstekst av geofysiske data ved Klimaameldingen"
Rapport 05/96 KLIBAS "KS - Spesifikasjon av programmer - Utskriftstramme"
Rapport 06/96 KLIBAS "KS - Utvikling av programmer: Døgnstramme, månedsramme, årsramme"
Rapport 08/96 KLIBAS "Brukerveiledning for: - Utskriftstrammer"
Rapport 13/96 KLIBAS "KLIBAS - status 30.06.1996"
Rapport 03/97 KLIBAS "Referater fra møter i databasegruppen 1996"
Rapport 07/97 KLIBAS "Arbeid gjort av Brukergruppa i 1996"
Rapport 45/97 KLIBAS "Logging av driftsparametre for kvalitetssikring av driftsutiner"
Rapport 46/97 KLIBAS "Kvalitetstrend QUAL_TREND i månedlig driftsrapportering"
Rapport 49/97 KLIBAS "Eksperiment med automatisk feilbehandling i KLIBAS"
Rapport 50/97 KLIBAS "Beregning og presentasjon av datakvalitet i månedlig driftsrapport"
Rapport 53/97 KLIBAS "Beregning og presentasjon av maskinvarekvalitet i månedlig driftsrapport"
Rapport 05/98 KLIMA "Kvalitetssikring av meteorologiske observasjonsdata"
Rapport 05/98 KLIBAS "Computer program SYNOP monitoring performance, quality and development of the TELE/SYNOP system"
Rapport 06/98 KLIBAS "Computer program CHECK_MAIL"
Rapport 07/98 KLIBAS "Arbeid i databasegruppen 1997"
Rapport 10/98 KLIBAS "Arbeid gjort av brukergruppa i 1997"
Rapport 13/98 KLIBAS "Analysing CRONTAB scheduling with computer program ORACLE_SHUTDOWN"
Rapport 14/98 KLIBAS "Automatically altering CRONTAB with computer program CHECK_15MIN v.2.0"
Rapport 15/98 KLIBAS "Computer program GRIM REAPER"