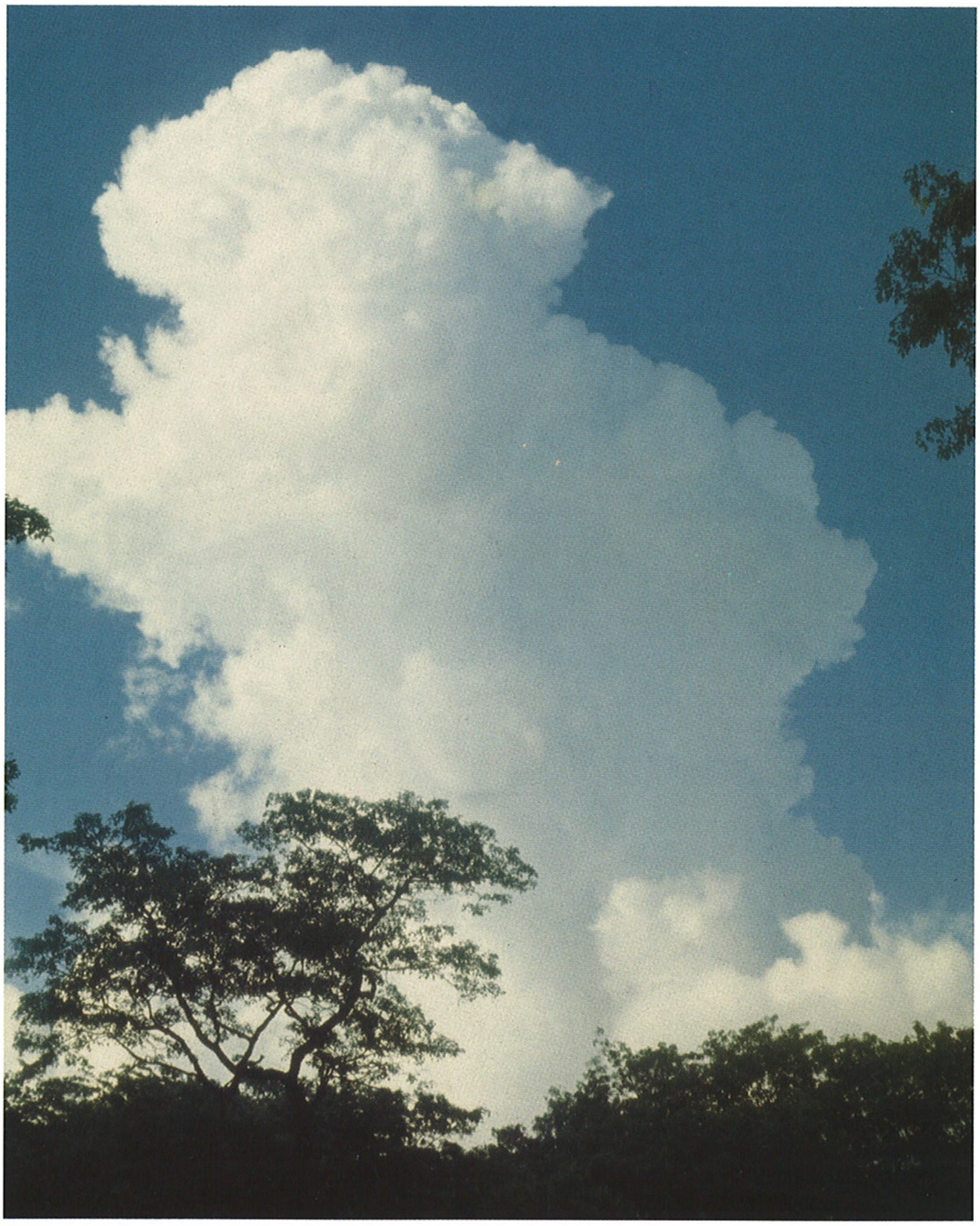
DNMI
Det norske meteorologiske institutt

KLIBAS research notes volume 2

Report no. 06/99

Petter Øgland (ed.)



KLIMA

DNMI - REPORT

NORWEGIAN METEOROLOGICAL INSTITUTE
BOX 43 BLINDERN N-0313 OSLO

PHONE: +47 22 96 30 00

ISSN 0805-9918

REPORT NO.

06/99 KLIMA

DATE

Feb 1 1999

TITLE

KLIBAS RESEARCH NOTES VOLUME 2

AUTHOR

Petter Øgland (ed.)

PROJECT CONTRACTOR

DNMI - Climatology Division / NORDKLIM

SUMMARY

In this volume 11 research notes dating from January 1999 are presented. All notes are related to on-going research and development of the KLIBAS climatological database system.

The problems analysed split into five categories.

1. General plans for research and development of the quality control systems in the KLIBAS climatological database systems.
2. Automatic weather stations (AWS).
3. The XVIND automatic weather stations (VIND_REG).
4. The TELE routine.
5. Software management.

KEYWORDS
1. Climatological databases
2. Meteorological data collection
3. Meteorological quality control
4. NORDKLIM, AWS, TELE, XVIND

SIGNATURE

....	....
Petter Øgland	Bjørn Aune
Research Scientist	Head of Climatology Division

Table of Contents

Foreword

This second volume of KLIBAS research notes contains a documentation of problems studied in January 1999. While most of the problems addressed in this collection of notes have later been solved, the notes only contain an analysis of the problem, sometimes with suggestions on how to solve it or alter the situation in order to eliminate the problem in its present form.

The notes are presented in order of writing. Five areas of climatological database research have been targeted for problem analysis.

1. *General plans for research and development of the quality control system in the KLIBAS climatological database system for the first half of 1999.* As it has been verified that DNMI will take part in the NORDKLIM Nordic climatolgical project, a summary of quality control achievments of 1998 with plans for further development during the first half of 1999 was written. Quality control is defined as activity no. 1.2 as a part of task 1, Climate Data, on the list of NORDKLIM tasks and activities [1].

2. *The automatic weather stations (AWS).* The first note contains a redesign of the AUTO_INN program, responsible for administrating the automatic data collection and quality control routines for the AWS. The redesign was implemented and documented in [2].

   The third note in this volume contains an analysis on how the monthly quality statistcs report is written and how the quality assurance group works, stressing differences and suggesting ways of improving on the report. Some of the suggestions have been put to use, as is apparent in [3].

   Another problem relating to quality control was the adaption of the "dead band" quality control method for running on the KLIBAS database system. Specification and discussion of the method is given in the sixth research note in this volume. The program itself is documented in [4] and tested on a dataset [5] before being a part of the daily control.

3. *The XVIND automatic weather stations (VIND_REG).* Several aspects of the XVIND routine were analysed during this month. The second note in this volume contains an analysis of what to do in the routine when non-defined stations appear on the files as it did on January 6th.

   On January 8th the VIND_REG program broke down again. The reason for this is analysed and documented in the fourth note in this volume. The analysis concluded with some suggestions for minor reprogramming in order to have the program fail in a more controlled manner if the type of problem causing the breakdown should ever occur again.

   For some reason the VIND_REG program also broke down when it could not identify a station at Vardø on January 19th. The reason for failure is analysed in the eighth note in this volume.

4. *The TELE routine.* One of the programs causing most troubles in the KLIBAS system is the SYNO_INN data collection program. As this is an extremely vital part of the KLIBAS system, an emergency crisis in the middle of the months was systematically analysed, concluding with suggestions on how to improve the program [6].

   Another problem that was addressed this month was observations in the TELE database table that either were missing, the interpolation programs apparently not working, or situations where the interpolation programs have inserted values for stations which did not observe this particular weather element. The analysis resulted in reprogramming of the SYNO_KONTR system [7].

   The reason why the interpolation programs did non-valid updates was then further analysed, resulting in the INTERPOL2 program being updated. As there are more problems down the pipeline for this program, an update of the system documentation waits for further revisions of the code.

5. *Software management.* The final note concerns itself with a problem suddenly occuring in the Computer Assisted Computer Programming (CACP) software developed at DNMI for making more systematic approaches to database development. The reason for the failure was found to be related to updates in the DRIFT program, which is under revision in order to fit as a monthly report on interpolation and quality control development related to the NORDKLIM quality control project [8].

The notes in this volume documents a number of problems in the KLIBAS database system during January 1999. The reason why these particular problems were being addressed this month had do to with the automatically updated priority list presented in the status report for December 1998 [9]. In the status report for January 1999 [8] more problems with dataflow, interpolation and quality control are illustrated.

Petter Øgland
Blindern, February 1st 1999

References:

[1] E. Førland, B. Dahlström, R. Heino, T. Jonsson and H. Madsen, *NORDKLIM. A NMD project on Nordic cooperation within climate activities.* DNMI/SMHI/FMI/VI/DMI, Oslo/Norrköping/Reykjavik/Copenhagen, 08.06.1998.

[2] P. Øgland, *Data transport and storage by program AUTO_INN v.1.2.* KLIBAS-report no. 01/99, DNMI, Oslo, 1999.

[3] P. Øgland, *AWS Quality Statistics December 1998.* KLIBAS-note no. 02/99, DNMI, Oslo, 1999.

[4] P. Øgland, *Testing for "dead band" by computer program ADK_REPEAT.* KLIBAS-report no. 04/99, DNMI, Oslo, 1999.

[5] Ø. Nordli and P. Øgland, *Comparison results for the new Glomfjord automatic station versus the present manually run station.* KLIMA-report no. 04/99, DNMI, Oslo, 1999.

[6] P. Øgland, *Reading data from syno-files into KLIBAS: SYNO_INN v.3.7.* KLIBAS-report no. 05/99, DNMI, Oslo, 1999.

[7] P. Øgland, *SYNO_KONTR v.2.1: Daily interpolation and quality control.* KLIBAS-report no. 06/99, DNMI, Oslo, 1999.

[8] P. Øgland, *Interpolation and Quality Control status for January 1999.* KLIBAS-note no. 03/99, DNMI, Oslo, 1999.

[9] P. Øgland, *KLIBAS Process Improvement December 1998.* KLIBAS-note no. 01/99, DNMI, Oslo, 1999.

Elaborating a warning system for internal problems in the AUTO_INN data transport and storage for Automatic Weather Station (AWS) data in KLIBAS at DNMI

Petter Øgland

Norwegian Meteorological Institute
January 5th, 1999

ABSTRACT

As the present AUTO_INN automatic weather station (AWS) data-transport and storage system at DNMI has a tendency of breaking down, causing AWS data to be unavailable for daily statistics for significant time of the month, systematic analysis in causes of breakdown is now being conducted. The present responsibility for the AUTO_INN program is discussed and new designs for the program is suggested.

AWS data collection by AUTO_INN

The AUTO_INN data transport and storage system was constructed in order to read AWS data from the "månadsfil"-format, or mnd-format as they are refered to here after, as described by Waage in his description of the AUTO data collection and formatting system [1].

The first version of AUTO_INN, called MND2ALA at the time, was completed in October 1994 and copied only data from the mnd-format files to the ALA data table in the Oracle RDBS KLIBAS. The system was revised in January 1995 (Øgland [3]).

In 1995 the inital programs were augmented into a system named ADI, the first version appearing in Mars 1995 [4] with a revision in May 1995 [5].

The present program, AUTO_INN was developed from the ADI system in June 1997 [6] and revised in July 1997 [7]. With the AUTO_INN program there has been a log system, updated since May 1997, containing the number of times the program has been run, how many times it has failed, how long time it takes on the average on each daily run and how many lines of code the program consists of.

During the past 18 month, the program has been run on the averate 35 times each month with a standard deviation of 11. In April 1998 it was run 56 times due to testing and improvement of the system. Figure 1 shows the relative number of runs that have failed.

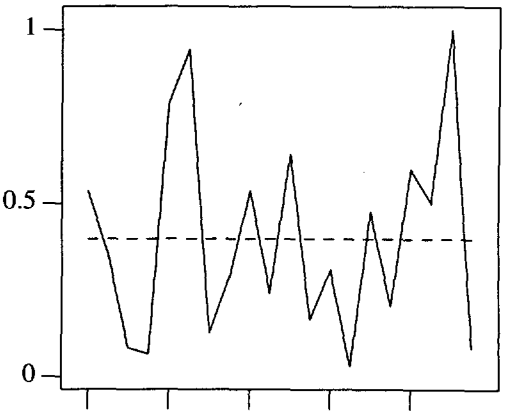


Fig 1. Relative number of executions to fail

In November 1997 AUTO_INN fail in 50 out of 53 cases, in December 1998 it fail in 33 of 33 cases. On the average the program fails in 40 out of 100 cases as is illustrated by the dashed line.

Reasons for failure

AUTO_INN is construced in a way so that it will fail whenever one of its subprograms fails. The main program makes system calls to five subprograms.

1. *MND2ALA* is reponsible for reading the mnd-files, checking these, inserting rows into ALA and HLA plus adjusting HLA if necessary and reporting problems to AUTO_INN.
2. *ADK* is the quality control module that checks observations in ALA.
3. *AUTO_MKK* is responsible for editing the results from ADK into daily reports that are automatically distributed by e-mail. AUTO_MKK is a simplified version of MKK that edits the ADK files into monthly reports.
4. *ALA2TELE* is reponsible for packing the AWS observations in ALA in SYNOP code an inserting then into TELE.
5. *ALA2ALV* has a similar function to ALA2TELE. In this case only observations fro Semi-Automatic Weather Stations (SAWS) are being used as only these are stored in ALV in addition to traditional weather stations. As ALA2ALV is run in batch mode from MND2ALA this is the only program that does not add warnings or errors to AUTO_INN.

The plot below shows the day-by-day number of automatically reported errors and warning during the last twelve months.

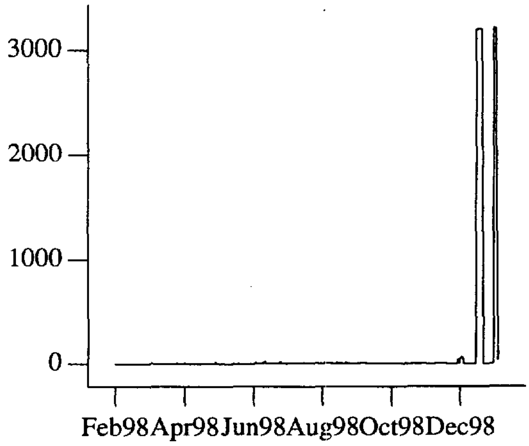


Fig 2. Time plot - problems day by day

A total of 29969 warnings and 72 errors have been recorded for the program AUTO_INN. The gross majority of the problems have been recorded during the past couplę of months.

Below is a histogram presenting the distribution of problems.

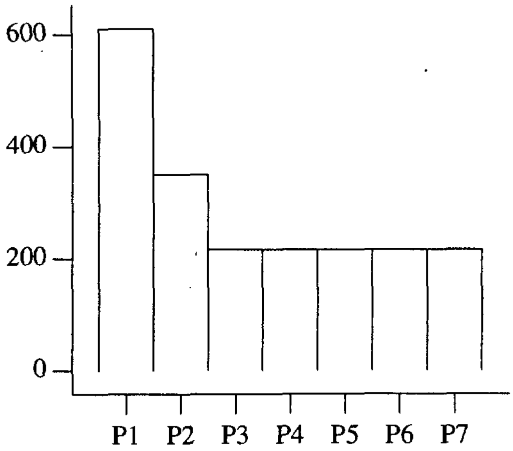


Fig 3. Frequency plot - 7 most critical problems

P1	Ppl=5553.5 => ppl:=NULL. *Count=610*
P2	Tnl=-273.1 => tnl:=NULL *Count=350*
P3	Stnr=99910, aar=1998, mnd=12, dag=7, ATX=6280 (ALA) *Count=216*
P4	Stnr=99910, aar=1998, mnd=12, dag=7, ATN=6280 (ALA) *Count=216*
P5	Stnr=99910, aar=1998, mnd=12, dag=7, ATT=6280 (ALA) *Count=216*
P6	Stnr=99910, aar=1998, mnd=12, dag=7, ATM=6280 (ALA) *Count=216*
P7	Stnr=99910, aar=1998, mnd=12, dag=3, ATN=6280 (ALA) *Count=216*

The total count of non-solved or non-ignored problems adds up to 30368. The list show the seven problems occuring the highest frequency amounting to 21 warnings (0 %). All these seven major problems are inherited from MND2ALA.

Suggested redesign for AUTO_INN

From figure 1 it seemed quite clear that AUTO_INN is not working very well in its present condition. When trying to analyse the reason for this further by a statistical analysis of the problems recorded so far, any significant data seem to be lost in a sea of information that is apparently only insightful for understanding how the MND2ALA program fails, not giving any light on the AUTO_INN program itself.

It seems imperative that the information flow in the AUTO_INN system is altered in a manner so that only problems relevant to AUTO_INN itself are logged on the AUTO_INN log. This information should then be necessary and sufficient for further analysis into which of the sub-programs that are causing the system to break down.

References

[1] Waage, E. *Brukarrettleiing AUTO. Datainnsamling frå automatiske vêrstasjonar.* KLIBAS-report no. 31/94, DNMI, Oslo, 1994.

[2] Øgland, P. *Månedlig rutine for innlasting av automatstasjonsdata i arbeidslager* KLIBAS-report no. 32/94, DNMI, Oslo, 1994.

[3] Øgland, P. *Innlasting og uthenting av automatstasjonsdata. Ny utgave.* KLIBAS-report no. 07/95, DNMI, Oslo, 1995.

[4] Øgland, P. *ADI: Automatisk datainnlasting for AUTO til arbeidslager* KLIBAS-report no. 13/95, DNMI, Oslo, 1995.

[5] Øgland, P. *Automatisk datainnlasting for AUTO med månedlig oppdatering av hovedlager: ADI v.1.1* KLIBAS-report no. 16/95, DNMI, Oslo, 1995.

[6] Øgland, P. *Døgnlig dataoverføring med AUTO_INN v.1.0 og AUTO2TELE v.2.0* KLIBAS-report no. 56/97, DNMI, Oslo, 1997.

[7] Øgland, P. *Innlesing AUTO_INN v.1.1 for AVS: Programmer mnd2ala, mnd2hla, ala2tele, adk og mkk* KLIBAS-report no. 59/97, DNMI, Oslo, 1997.

Elaborating a scheme for handling wrong references in data files for automatic wind sensors (VIND_REG) at DNMI

Petter Øgland

Norwegian Meteorological Institute

January 6th, 1999

ABSTRACT

The VIND_REG data transport program is responsible for reading automatic wind registrations from data files and inserting these into the VINDX Oracle data table in the KLIBAS database system. In some of the files it happens that the identification given by the name of the file does not correspond with the identification inside the file. This note discusses problems assositated with this and makes suggestions in order to improve the situation.

The VIND_REG program

Specifications for the VIND_REG program were written by Lars Andresen in September 1997, resulting in a first version of VIND_REG being released the 12th of July 1998 [1]. The purpose of the program was to read files, interpret and modify observations when necessary and store the results in suitable datatables in the KLIBAS database.

By August 18th 1998 a routine was being established at the Climatology Division for handling the observations. A quality control program VINDDEK was then realeased according to specifications by Lars Andresen and P.O. Kjensli [2].

By the end of August the same year, a revised version of VIND_REG appeared [3]. Minor faults in the algorithm had been corrected, the specification had been slightly updated.

Still there were problems with the program, and on the 20th of October 1998 a version 1.2 of VIND_REG was released [4]. The program was now running systemaically every hour of the day by the crontab schedule on the UNIX machine. A new table VINDX was designed for storing the observations.

The first log of the program is in April 1998. It was then run 17 times, failing 0 of these (0.0 %). The number of users during this first month was 1. The curve below shows the relative number of runs of the program that has been failing according to the log.

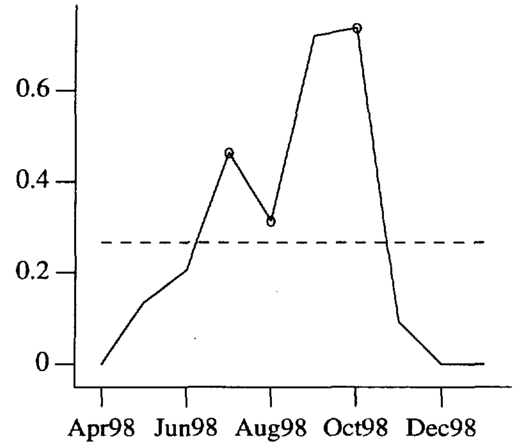


Fig 1. Relative number of failures

The average value dashed in the figure no 1. is 0.2668. The time when the revisions of the program appeared are marked with a small circle.

The identification problem

In the middle of November 1998 a vast number of problems were recorded for a file who was by filename identified as ENSYT (Sortland), but inside the file stated to be ENRY (Rygge).

The same problem had for previous months been recorded for ENNR (Narvik).

In order to identify such problems, the VIND_REG program generates warnings whenever such a happening occurs. This has resulted in an enormous output of warnings for the VIND_REG program.

As a result of discussing what to do with these warnings with P.O. Kjensli and L. Andresen medio November, the manual routine was further formalised by adding a function for checking the files before the VIND_REG program manages to read the observations. A note was also sent to Henrik W. Nilsen of the Instrument Division in order to secure that no further incidences would occur.

Conclusion

In principle this type of problems should not reoccur. If, however, they did reoccur, the routine should be notified. No reprogramming of VIND_REG is presently required.

References

[1] P. Øgland, *Computer program VIND_REG.* KLIBAS-report no. 22/98, DNMI, Oslo, 1998.

[2] P. Øgland, *Computer program VINDDEK.* KLIBAS-report no. 39/98, DNMI, Oslo, 1998.

[3] P. Øgland, *Computer program VIND_REG v.1.1.* KLIBAS-report no. 43/98, DNMI, Oslo, 1998.

[4] P. Øgland, *Storing AWS wind observations by use of VIND_REG v.1.2.* KLIBAS-report no. 56/98, DNMI, Oslo, 1998.

Differences between how information is presented in the monthly AWS quality statistics report and how the AWS quality assurance group works

Petter Øgland

Norwegian Meteorological Institute
January 7th, 1999

ABSTRACT

In order to maintain and improve data quality for automatic weather stations (AWS) at DNMI, the Climatology Division and Instrument Division meet on monthly basis to discuss problems related to quality of AWS. Discussions are, to a great extent, based on statistical material provided by the Climatology Division by the monthly AWS quality statistics report. Although the monthly report contains important input for discussion, present methods of displaying and organising the statistics are not always in cor respondance with how the meetings are conducted. In this note both the meeting and the report are discussed, suggesting ways to improve the quality of both.

AWS Quality Assurance

In April 1992 a general specification of automatic weather stations for DNMI was written [1]. The series of monthly statistical quality assurance (QA) reports that have been written since February 1995 [2] have displayed the number of irregularities that have been found between the specification and what have generally been produced at each station each consecutive month.

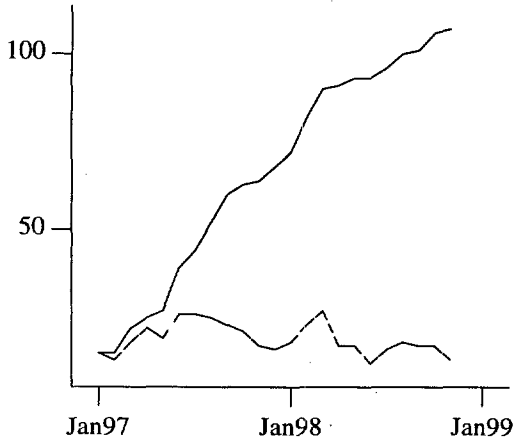


Fig 1. Problems discussed by the AWS group

In January 1996 it was decided by the Climatology Division and the Instrument Division [3] that there should be carried out monthly meetings where the two could join and discuss problems relating to quality and quality improvement.

An average of 4 problems have been processed by the QA group every month. The average number of problems on the agenda every month is 19.

A summary of the collected works of the QA group for 1996 is documented in [3]. Collected works of 1997 is documented in [4]. Figure 1 displays the total number of problems discussed by the group (solid) and the number of problems resolved each month (dashed).

Collecting monthly AWS quality statistics

During one of the first days of each month, a quality statistics report for the previous month is produced. The report usually gets authorised, published and distributed less than a week after being written.

Date for the upcoming QA meeting is the verified from summary of last meeting and distributed by e-mail. If at least three of the four people presently definig the group can meet, a meeting is held.

The meeting consists of two parts. First the summary of last meeting is used as agenda in order to check that every problem from last month have been solved. Then additional problems are discussed. The monthly statistics report is used during both parts of the meeting, often more vital for the second part.

The QA report is organised in four chapters. The first chapter gives a general point of view on quality, consisting of month to month development of reported problems and distribution of problems among stations for the present month and the last 12 months.

Chapter two is then a description of the AUTO data collection system and the ADK quality control system used for measuring quality.

Chapter three contains a station by station statistical description of quality, while chapter four is a summary of all e-mail messages distributed that have relevance for the quality statistics.

Differences in approach

During the first part of the meeting, when old issues are debated, the order of problems follow the summary which is usually printed in an appendix to the QA report.

There are many differences in the QA approach of the meeting and the approach given by the report. By investigating the latest KLIBAS-note, AWS Quality Statistics November 1998 [9], a list of six problems are especially noted.

1. There is no systematic correspondance between the agenda given by the summary of last meeting and the structure of the QA report.
2. In order to give the QA analysis some structure, at the meetings one sometimes have to distiguish between general problems and specific problems. As the general problems often are of greater importance, a software problem perhaps distoring data on a whole set of stations, or communication problems making the whole system break down, general problems have to be distinguishable from specific problems such as problems with temperature measurements or global radiation measurements at a particular station. In

   the present report they are not clearly distinguished.
3. In the report the AWS are presented one by one, sorted by identification number and regardless of category. The main interest of the QA group, however, is to investigate the stations run by DNMI. Other stations may provide interesting information, but for none of these is it possible for DNMI to inspect the stations and solve problems.
4. In the report there is a quality index that is a mixture between data completeness and the number of errors registered at a station. When discussing stations at the meetings, completeness and errors are seperate issues.
5. When, at a meeting, a problem is found at a certain station, it is often interesting to ask whether this problem also occurs on other stations. In the report there is no analysis of this kind to support this type of investigation.
6. Sometimes it happens that, due to various reasons, a certain AWS is missing from the statistics, or is represented by the wrong identification number. In [9] an ITAS station has in fact been lost due to renumber of identification numbers at DNMI. If the QA meeting were to discuss this particular station, the report is of no help.

Conclusions

Each of the issues discussed above show instances where the correspondance between how the QA meeting is conducted and how the report is written could have been better. In order to improve and sustain high quality for the AWS changes should indeed be performed in order to improve on the situation.

Below are suggestions for each particular problem is given below.

Problem 1. The report should be redefined in order to function as a agenda for the meetings. The meeting summaries should also be adjusted inheriting the same structure as the meeting and the report.

Problem 2. The the differenciation between general and spesific problems should be made clear. Unlike the specific problems that are already linked up with the stations, there is no classification of the general problems in the QA report or the QA meeting so far.

Some general problems that often crop up do, however, seem to fall into two major categories, problems having to do with collecting data and problems having to do with quality of data being collected. The category of problems having to do with data collection may then be divided into groups of three.

A. *Lack of weather sensors.* This is a common problem, as examplified in at least two cases above. The problem is usually solved by inspections where weather sensors are mounted.

B. *Communication problems.* Communication has been discussed in the case of Sognefjell.

C. *General lack of data in the database.* This is the case if the AUTO_INN program at the Climatology Division has borken down, or the AUTO system at the IT Division has failed in some way, if the communication between DNMI and a group of stations has broken down or a group of stations has broken down. The first case is generally not commented upon as this is a local problem.

The category of problems having to do with quality of data being collected can also be devided into three groups:

D. *Sensor problems.* Sometimes a sensor or a series of sensors may be wrong. In some cases such sensors have been tested at the Blindern test lot. Sometimes they have been tested in action.

E. *Station software problems.* Numerous software poblems have been discussed and analysed in 1996-1998. Some have been corrected.

F. *Problems with the quality control system.* The quality control system ADK has been revised a number of times in order to mark only observations that are obviosly wrong. In this case measured data quality is misguiding and should generally not be used for decision making before being corrected.

Problem 3. In order to not to mix up different types of stations, the report should distinguish in a clear way between the different types. In the present statistical quality report [9] there are AWS data collected from four different sources:

1 *DNMI AWS.* There are at the moment 40 stations producing observations for storage in the KLIBAS database system. Of these 27 are of Scan-Matic type, 7 of Vaisala/Milos type and and 6 of Campbell type. The set of sensors varies from station to station.

2 *ITAS AWS.* This is a group of 10 stations of Campbell type with a fixed set of weather sensors producing weather measurements of wind, air temperature, ground temperature at five levels, relative humidity, precipitation and others.

3 *Miljødatasenteret AWS.* 5 oil rigg automatic weather stations make up this group. Stations are made by Aanderaa Instruments. All stations have the same set-up of parametres: wind (FF, DD, FG), air pressure (P, P0), air temperature (TT, TN, TX) and relative humidity (UU).

4 *Vegvesen AWS.* The group consist of 4 stations. Neither of them are designed in order to correspond with DNMI specifications and maintenance is done independantly of DNMI. The stations are defined as test stations in the KLIBAS database and are at the present not included among the stations in the quality statistics report.

Problem 4. In order to make clear distinctions between data completeness and data errors, the report should begin by comparing data completeness for all stations. The combined data completeness and data problems should then be discuss when handling the stations seperately.

Problem 5. In order to make it possible to investigate whether a problem at a particular station is a part of a generall problem for more than one stations, a description of data quality for each group of weather sensors for each group of stations should be given.

Problem 6. In order to prevent that stations are missing or wrongly identified, the program that is responsible for producing the monthly report should contain checks that make the program terminate in error when such problems arise.

General conclusions. If the suggestions in this article prove to be worthwhile, this should be reflected in the continuation of the curves in figure 1. Both curve representing the number of problems discussed at each meeting and the curve showing the accumulated number of problems being solved by the group should increase, the latter should increase by a generally larger growth rate.

A new investigation should be carried out in another 12 months time in order to check if the process has improved. As an alternative, the monthly report could contain statistics that evaluated the work of the group month by month.

References

[1] P.O. Kjensli et al., *General Specification of Automatic Weather Stations for DNMI* DNMI, internal note, Oslo, April 1992.

[2] P. Øgland, *Kvalitetskontroll av automatstasjonsdata februar 1995.* KLIMA-report no. 12/95, DNMI, Oslo, 1995.

[3] P. Øgland, R. Brækkan, S.L. Lystad, L.G. Olonkin, *Arbeidsgruppe for kvalitetssikring av automatiske værstasjoner 1996.* KLIBAS-report no. 21/96, DNMI, Oslo, 1996.

[4] P. Øgland, *AUTO system statistics.* KLIBAS-note no. 19/98, DNMI, Oslo, 1998.

[5] P. Øgland, *AWS Quality Statistics October 1998.* KLIBAS-report no. 25/98, DNMI, Oslo, 1998.

[6] P. Øgland, *Kvalitetskontroll av automatstasjonsdata oktober 1997.* KLIBAS-report no. 21/97, DNMI, Oslo, 1997.

[7] P. Øgland, *Kvalitetskontroll av automatstasjonsdata november 1997.* KLIBAS-report no. 23/97, DNMI, Oslo, 1997.

[8] S.L. Lystad and P. Øgland, *ADK/ADL: Datakontroll for EDAS automatstasjoner.* KLIBAS-report no. 36/94, DNMI, Oslo, 1994.

[9] P. Øgland, *AWS Quality Statistics November 1988.* KLIBAS-note no. 27/98, DNMI, Oslo, 1998.

Improving robustness for the AERO stations routine (VIND_REG) at DNMI by eliminating an error caused by a recent data file from Hammerfest Lufthavn

Petter Øgland

Norwegian Meteorological Institute
January 11th, 1999

ABSTRACT

The VIND_REG data transport program is responsible for reading automatic wind registrations from data files and inserting these into the VINDX Oracle data table in the KLIBAS database system. The system is not totally robust, as became evident on January 8th 1999 when VIND_REG collapsed due to unexpected content on a data file from Hammerfest. This note gives an analysis of this particular problem and suggests how the VIND_REG program may be improved in order to prevent similar problems from reoccuring.

The VIND_REG program

Specifications for the VIND_REG program were written by Lars Andresen in September 1997, resulting in a first version of VIND_REG being released the 12th of July 1998 [1]. The purpose of the program was to read files, interpret and modify observations when necessary and store the results in suitable datatables in the KLIBAS database.

By August 18th 1998 a routine was being established at the Climatology Division for handling the observations. A quality control program VINDDEK was then realeased according to specifications by Lars Andresen and P.O. Kjensli [2].

By the end of August the same year, a revised version of VIND_REG appeared [3]. Minor faults in the algorithm had been corrected, the specification had been slightly updated.

Still there were problems with the program, and on the 20th of October 1998 a version 1.2 of VIND_REG was released [4]. The program was now running systemaically every hour of the day by the crontab schedule on the UNIX machine. A new table VINDX was designed for storing the observations.

The first log of the program is in April 1998. It was then run 17 times, failing 0 of these. The curve below shows the relative number of runs of the program that has been failing according to the log.

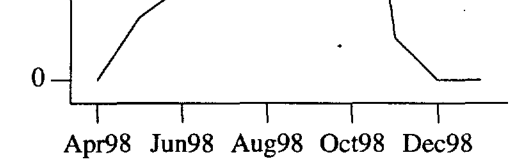


Fig 1. Relative number of failures

The average value dashed in the figure no 1. is 0.2668. The time when the revisions of the program appeared are marked with a small circle.

The Hammerfest problem

On January the 8th 1999 the VIND_REG program failed, reporting the error "Exit status from vind_reg indicates error" which by closer analysis was due to a segmentation fault in the system. This is a serious error meaning that the termination of the program was not controlled.

By testing and experimenting it turns out that the reason for program failure is located on the final line of the input file. The two final observations on the Hammerfest file are dated 21:00 and 21:22. In the function "ObsInit" this seems to be acknowledged, but in the procedure "dataManipulation" the program acts as if the final observation was done later than 22:00 with no observation at 22:00, consequently trying to construct the 22:00 observation from previous recordings during the last sixty minute s and update the constructed observation with flag '4'.

The program collapses in the update function "setFLBRUDDeq4" when trying to do the assignment "j=obs->next->index" where next is a pointer to a structure that should normally contain space for the integer variable "index" but which in this case is not defined (NULL).

Now, why did this problem occur in the first place? The "setFLBRUDDeq4" function was called from the "dataManipulation" function. The criteria for adressing "setFLBRUDDeq4" was here that the final observation at 22:00, temporarily created by the program itself, contained information saying that after the previous observation time, 21:00, there was at least one fracture and the index value for the 22:00 observation was negative.

The index value is an internal value of the program used for indicating whether the observation is read from file or missing. A negative value, such as in this case, would indicate that the observation was missing. Had it not been for the fact that this was the very last observation on the file, it would indeed have been correct to adress the "setFLBRUDDeq4" function in order to fill in for the 22:00 observation, using fracture information from the past 60 minutes.

In this case, however, as the last observation on file was at 21:22, creating a 22:00 observation would probably be wrong as the next file from Hammerfest would perhaps include real data for this hour of observation.

What the program should have done then, would be store the 21:22 observation and wait for next file to do updates on the 22:00 observation.

Conclusion

First VIND_REG should be reprogrammed so that it will break down in a controlled manner in order to prevent using unnecessary long time in order to locate an error of this type.

Second, the breakpoint in "dataManipulation" where "setFLBRUDDeq4" should be modified so that the condition only works for observations that are not the last one on file.

References

[1] P. Øgland, *Computer program VIND_REG.* KLIBAS-report no. 22/98, DNMI, Oslo, 1998.

[2] P. Øgland, *Computer program VINDDEK.* KLIBAS-report no. 39/98, DNMI, Oslo, 1998.

[3] P. Øgland, *Computer program VIND_REG v.1.1.* KLIBAS-report no. 43/98, DNMI, Oslo, 1998.

[4] P. Øgland, *Storing AWS wind observations by use of VIND_REG v.1.2.* KLIBAS-report no. 56/98, DNMI, Oslo, 1998.

Planning research and development of the quality control system in the KLIBAS climatological database system for the first half of 1999

Petter Øgland

Norwegian Meteorological Institute

January 12th, 1999

ABSTRACT

Apart from general problems assosicated with the KLIBAS database system, problems having to do with handling the Oracle database core or hardware problems, eight specific areas of climatological database research and development are addressed. These are the TELE routine, the METAR routine, the ALV weather station routine, the PIO routine, the AERO routine, the real-time AWS routine, the Aanderaa AWS routine and the ALN precipitation routine. In this note a few comments on status and shortcomings of each of these are made and tentative plans for research and development for the first half of 1999.

Introduction

A major problem inherit in the KLIBAS research and development is how to chose the best problems to solve. At the time of writing there have been recorded problems in 113 of the 131 computer programs (86%) that are logged by the system.

From this pool of problems, how does one chose the most important ones to solve? The method used since January 1998 has been the "Pareto strategy" of statistical quality control that consists of sorting problems according to how many warnings are being generated and systematically select programming jobs from the top of this list [1].

In order to estimate how much work expected to be done during the first half of 1999, a look at statistics from previous year may give an indication.

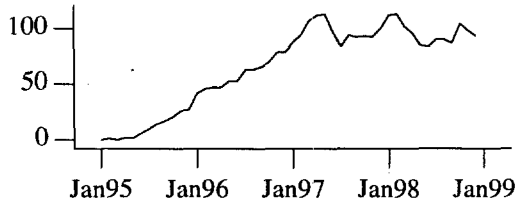


Fig 1. Number of running programs in KLIBAS

The curve in figure 1, showing the number of programs running for each consequtive month, seem to indicate that while there was a rapid development of new programs in 1995 and 1996, in 1997 and 1998 most time has been spent on maintaining the approx. 100 programs that now make up KLIBAS.

Hopefully the systematic approach to programming, that has been applied with increasing persitency since 1995, should have made the present group of KLIBAS programs mature enough during 1997 and 1998 as to pave way for a new growth of programs in 1999.

As a general guide line, however, in most of the routines that will be discussed below, there is a great need for quality control software [2].

The TELE routine

Data in the TELE data table is used for daily weather statistics, reading observations for the last 30 days, STATUT [3], it is used for daily updates of weather curves on the internet, SYNO_ESPEN [4], it is used for monthly climatological statistics KA_H_STAT [5], and it is used for weekly statistics, VSUKE [6]. A more detailed overview of the programs and the TELE/SYNOP climatological database system is given in [7].

If the TELE programs are to produce meaningful weather statistics, it is imperative that the quality of the observations is acceptable. Checking the content of the TELE table is a manual responsibility, aided by the S-T-F quality control computer programs and some automatic checks and convertions that are being done when observations are being inserted into the TELE table.

The approach to quality control of 1998 was to make programs CHECK_H_STAT [5] and CHECK_STATUT [8] that performed an analysis of the output from the central KA_H_STAT and STATUT programs. The main idea behind this approach is to simulate on a computer the kind of analysis that is normally done manually, checking columns in the output and cross-checking the TELE table and other statistics where this is the natural thing to do.

During 1998, however, most of the time spent on the TELE routine had to be devoted to other problems. The SYNO_INN program was revised 4 times, the latest version 3.6 released in November [9]. The INTERPOL2 program was revised, and a new interpolation program INTERPOL_P0 was established [10] and later revised [11]. An improved version 2.0 of the S-T-F quality control program was also relased [12].

Additional time was spent constructing a data flow from the metar data table ALF to TELE [13] and on improving the VNN part of the TELE routine [14].

Present priority lists indicate that more work on the interpolation programs will be performed before more programming will be done on the CHECK_STATUT and CHECK_H_STAT programs.

The METAR routine

The metar observations stored in the METAR data table and now also in the ALF data table are used primarily for reference data in the manual quality control of ALV and TELE observations.

There is no quality control for metar data stored in the KLIBAS data tables except some minor checks in the META_INN data collection program [15]. During 1998 a dataflow from the ALF table into the TELE data table was constructed [13], and this may be a natural point of starting a quality control for the metar data by using the TELE quality control programs.

The ALV weather station routine

In the ALV weather station routine quality control programs consist of the CONTSYN programs that have survived more or less intact through several database changes since the early 1970's [16].

The same approach to improving the quality control procedures as in the TELE routine were tested also with the ALV routine. In this case a simulation program KLIMA_KONTR was constructed [17] which would run the CONTSYN quality control programs in the same order as they were being run manually, KLIMA_KONTR collecting statistics from the different programs in order to measure quality.

Special programs CHECK_RELFUKT [18] and CHECK_KONTHUM [19] were developed for analysing particular modules in the CONTSYN system. The CONTSYN1 program, responsible for testing completeness, was totally reprogrammed from scratch [20].

As semi-automatic weather stations (SAWS) also became a part of the observations that were to be handled by the ALV routine, a program ALA2ALV was construted in order to update SAWS observations in ALV with values from ALA [21].

Improvement of the quality control routines for the ALV routines will have partly to consist of reprogramming of the CONTSYN programs and partly on developing the KLIMA_KONTR ideas further.

The PIO routine

In May 1998 a new type of stations were introduced at DNMI, the "PC in the Observations services" or PIO stations. A program PIO_INN for reading observations of this type into the KLIBAS database system was established and revised several times until a workable solution was running [22].

There are no specific plans for a quality control for the PIO stations outside the PIO_INN program although work has commenced on a PIO_KONTR program that will help verifying that the data collection is working properly.

If the PIO stations will be inserted in the ALV routine, quality control programs for the ALV routine will naturally be used for quality control of the PIO observations.

The AERO routine

In July 1998 the first version of the VIND_REG program for reading automatically generated wind observations (AERO stations) was established [23]. The program was later revised, version 1.2 is presently running [24].

As a routine was established, a quality control VINDDEK was created [25]. Work on a more sophisticated quality control program, VIND_KONTR, has commenced.

The ALA automatic weather station routine

Observations from automatic weather stations (AWS) are presently used for updating the TELE [26] and ALV [21] tables and also for producing statistics DOGNEKSTREMRAMME, AARSRAMME, AARSREKKERAMME, MNDRAMME and VINDFREK [27]. During 1998 the programs MNDTIMRAMME [28], AARSTIMESRAMME [29] and a revised version of AARSRAMME [30] were added to the system.

A serious defect of the system is that there are no systematic updates of the ALA and HLA as a result of the quality control. Progress in quality control for AWS during 1998 consisted of improvement of the error statistics programs FEILOVERSIKT [31] and revisions of the MKK program [32] and ADK program [33].

The ALA routine has in 1998 been jammed by a series of problems having to do with data collection AUTO_INN [34] and backup by AUTO_BACKUP [35].

Software for updating the ALA and HLA tables should be developed in parallell with the AANDERAA automatic weather station routine.

The AANDERAA automatic weather station routine

As with the ALA routine, the AANDERAA routine also lacks vital quality control programs.

During 1998, however, no work has been done in order to improve upon this, although more work will hopefulle be done during the first half of 1999, problems ranking high on the "objective" priority list mentioned above.

Software for updating the AANDERAA and HLA tables should be developed in parallell with the ALA automatic weather station routine.

The ALN precipitation station routine

There is a significant lack of good quality control programs in the ALN routine, but just as in the case of the AANDERAA routine, little work was done in order to improve the situation during 1998, with the exception of a revised version of the RRUTM precipitation matrix program was released [36].

For the ALN routine there are, however, several quality control programs that have been under development but not yet surfaced in operative versions. These programs include an areal check ROMRR, a general quality control system FREYR and a quality control simulation and evaluation program PRECIP_KONTR, all sketchly described in [37].

Final remarks

A considerable effort has been put in during 1998 in order to improve the KLIBAS database system. The improvement rate of the system, however, does not seem to imply that one should expect fully operative quality control routines for all eight systems within the next six months.

Nevertheless, it should be fair to expect some improvement in most of the routines, the ALV, ALA and ALN being perhaps of special significance in terms of being generators of the most important raw material used by the Climatological Division.

it is, however, the general robustness of the KLIBAS system that makes the priorities. The priority list is automatically updated so that the programs causing the most problems are handled first. Hopefully the break in program growth in 1997 and 1998, indicated in figure 1, may have caused the KLIBAS system to mature in order to bring new growth.

References

[1] P. Øgland, *Towards an ISO-9000 standard for system development by DRIFT.* KLIBAS-report no. 47/98, DNMI, Oslo, 1998.

[2] L. Andresen et al., *Kvalitetssikring av meteorologiske observasjonsdata.* KLIMA-report no. 06/98, DNMI, Oslo, 1998.

[3] P. Øgland, *DAGLIGSTATISTIKK - Programmet statut v.1.2.* KLIBAS-report no. 57/97, DNMI, Oslo, 1997.

[4] P. Øgland, *Feeding observations from TELE to VA by SYNO_ESPEN v.2.1.* KLIBAS-

report no. 48/98, DNMI, Oslo, 1998.

[5] P. Øgland, *Computer program CHECK_H_STAT for monitoring missing values on KA_H_STAT.* KLIBAS-report no. 18/98, DNMI, Oslo, 1998.

[6] P. Øgland, *Ukesoversikter fra TELE med programmet VSUKE v.2.0.* KLIBAS-report no. 48/97, DNMI, Oslo, 1997.

[7] P. Øgland, *The TELE/SYNOP Climatological Database System.* KLIMA-report no. 07/98, DNMI, Oslo, 1998.

[8] P. Øgland, *Computer program CHECK_STATUT v.1.1.* KLIBAS-report no. 35/98, DNMI, Oslo, 1998.

[9] P. Øgland, *Reading data from syno-files into KLIBAS: SYNO_INN v.3.6.* KLIBAS-report no. 65/98, DNMI, Oslo, 1998.

[10] P. Øgland, *Interpolating air pressure in TELE with INTERPOL_P0.* KLIBAS-report no. 46/98, DNMI, Oslo, 1998.

[11] P. Øgland, *Interpolating air pressure in TELE with INTERPOL_P0 v.1.1.* KLIBAS-report no. 52/98, DNMI, Oslo, 1998.

[12] P. Øgland, *Quality control computer program S-T-F v.2.1: Automatic removal of "bad" INTERPOL2 values.* KLIBAS-report no. 17/98, DNMI, Oslo, 1998.

[13] P. Øgland, *Updating METAR from ALF in TELE by ALF2TELE.* KLIBAS-report no. 60/98, DNMI, Oslo, 1998.

[14] P. Øgland, *Updating data in TELE by VNN v.2.1.* KLIBAS-report no. 62/98, DNMI, Oslo, 1998.

[15] P. Øgland, *Reading METAR from meta-files into ALF by META_INN v.1.1.* KLIBAS-report no. 57/98, DNMI, Oslo, 1998.

[16] L. Håland and P. Øgland, *Kvalitetskontroll av værstasjonsdata i Klimaavdelingen.* KLIBAS-report no. 23/94, DNMI, 1994.

[17] P. Øgland, *KLIMA_KONTR: Simulation and control of a quality control system for weather data.* KLIBAS-report no. 24/98, DNMI, Oslo, 1998.

[18] P. Øgland, *Computer program CHECK_RELFUKT.* KLIBAS-report no. 37/98, DNMI, Oslo, 1998.

[19] P. Øgland, *Analysing Relative Humidity Statistics by CHECK_KONTHUM.* KLIBAS-report no. 38/98, DNMI, Oslo, 1998.

[20] P. Øgland, *Test for completeness in data series by CONTSYN1 v.3.0.* KLIBAS-report no. 69/98, DNMI, Oslo, 1998.

[21] P. Øgland, *Updating AWS in ALV and TELE by ALA2ALV v.1.2.* KLIBAS-report no. 66/98, DNMI, Oslo, 1998.

[22] P. Øgland, *PC i observasjonstjenesten: PIO_INN v.2.2.* KLIBAS-report no. 58/98, DNMI, Oslo, 1998.

[23] P. Øgland, *Computer program VIND_REG.* KLIBAS-report no. 22/98, DNMI, Oslo, 1998.

[24] P. Øgland, *Storing AWS wind observations by use of VIND_REG v.1.2.* KLIBAS-report no. 56/98, DNMI, Oslo, 1998.

[25] P. Øgland, *Computer program VINDDEK.* KLIBAS-report no. 39/98, DNMI, Oslo, 1998.

[26] P. Øgland, *Innlesing AUTO_INN v.1.1 for AVS: Programmer mnd2ala, mnd2hla, ala2tele, adk og mkk.* KLIBAS-report no. 59/97, DNMI, Oslo, 1998.

[27] P. Øgland, *Utskriftsrammer for verifikasjon og testutskrift av AVS-data: Døgnekstremramme, årsramme, månedsramme, årsrekke-ramme og to-parameter-ramme.* KLIBAS-report no. 16/96, DNMI, Oslo, 1996.

[28] P. Øgland, *Computer program MNDTIMRAMME.* KLIBAS-report no. 31/98, DNMI, Oslo, 1998.

[29] P. Øgland, *Dumping hourly wind observations to file by AARSTIMESRAMME.* KLIBAS-report no. 50/98, DNMI, Oslo, 1998.

[30] P. Øgland, *Producing daily AWS averages and sums by AARSRAMME v. 1.1.* KLIBAS-report no. 64/98,, DNMI, Oslo, 1998.

[31] P. Øgland, *AWS Computer programme FEILOVERSIKT. KLIBAS-report no. 02/98, DNMI, Oslo, 1998.*

[32] P. Øgland, *Monthly AWS Quality Control Statistics MKK v.2.0.* KLIBAS-report no. 49/98, DNMI, Oslo, 1998.

[33] P. Øgland, *AWS real-time quality control: ADK v.2.2.* KLIBAS-report no. 55/98, DNMI, Oslo, 1998.

[34] P. Øgland, *Data transport and storage by program AUTO_INN v.1.2.* KLIBAS-report

no. 01/99, DNMI, Oslo, 1999.

[35] P. Øgland, *Daily export and import of A-tables by AUTO_BACKUP v.2.1.* KLIBAS-report no. 61/98, DNMI, Oslo, 1998.

[36] P. Øgland, *Precipitation matrix program RRUTM v.3.0.* KLIBAS-report no. 12/98, DNMI, Oslo, 1998.

[37] P. Øgland, *PRECIP system statistics.* KLIBAS-note no. 18/98, DNMI, Oslo, 1998.

Adding a "dead band" test to the ADK quality control system for automatic weather stations at DNMI

Petter Øgland
Øyvind Nordli

Norwegian Meteorological Institute

January 14th, 1999

ABSTRACT

In order to identify if a weather sensor has blanked out, a "dead band" test chould be introduced to the ADK quality control system. In this report, the "dead band" test is discussed in various forms, suggesting how the test should be implemented in order to fit the ADK quality control system at DNMI.

Introduction

The introduction of a "dead band" test in the ADK system has been motivated by two major quality assurance problems recorded in December 1998, sensors for relative humidity need to be changed, and wind sensors need to be changed [1].

The term "testing of dead band" has been used by the Swiss Meteorological Institute [2] for describing a test for identifying time series that do not vary as much as expected, giving grass temperature sensor (5 cm) covered by snow as an example.

Other examples given by the Swiss Meteorological Insitute are: frozen wind instruments, frozen weather hut and damaged temperature instrument. Data are suggested to be corrected manually.

The "dead band" test is also know under other names. At DNMI the test has been called "lik verdi-kontroll (repeated value control)" [3]. The program to be specified in this note is, however, given the name ADK_REPEAT.

Interfacing with the ADK system

The specifications for the ADK system are given in [4]. It was implemented to be run on KLIBAS in December 1994 [5], with later revisions in Mars 1995, version 2.0 [6], July 1997, version 2.1 [7] and October 1998, version 2.2 [8].

The ADK system is designed by modules, using different subprograms to perform different tasks. Adding a new type of checks to the system should then, in principle, not be severely difficult, although the interface for the ADK program need to be specified, i.e. the results produced by the program should fit into the ADK in such a manner that it is compatible with the MKK [9] and AUTO_MKK [7] error statistics programs.

Making the ADK_REPEAT compatible with MKK and AUTO_MKK should, on the other hand, not amount to more than having the program updating ascii files that may later be read by the MKK and AUTO_MKK programs.

How long data periods should be tested?

For short term quality control, it may be sufficient to check for the past thirty days. The MKK statistics produced by AUTO_MKK give daily statistics, while the MKK program gives monthly error profiles of the stations. In the case of "dead band" tests, however, there is no guarantee that the signal has been dead for over a month. Indeed, for some automatic weather stations, such as the DSU-based Aanderaa stations, data is sometimes collected with annual intervals, and "dead band" tests should be made to apply for at least one year.

In order to prevent the program from reading all too much data into the local memory, log files should be used, so the program can be run iteratedly on a set of observations, reading only about one month (the last 30 days) at the time, but updating the files so the complete period of dead signal is recorded.

Which AWS sensors should be checked?

In principle all sensors should be checked, although a selection may be implemented for the first version of the program. In order to check the program, the snow depth sensor at Glomfjord is known to be dead and should make good test data [10]. Apart from this, checks for relative humidity and wind are of great importance as testing of these elements was the motivation of the quality control program in the first place.

How to chose parameters for the test

The repeated value test that is the basis for [1] used about three observations (hours) in order to notify that the sensor may be faulty. This test was, however, used for maritime data.

When testing snow depth or relative humidity on non-maritime automatic weather stations there may be days without change, say, two or three days where relative humidity is constantly equal to 100%. In some cases, such as with the snow depth described in [10], there are several months without change, and these cases are the ones that should be addressed at first before trying to refine the technique.

Before attempting to calibrate the test, 120 consecutive observations (5 days) may be applied as an initial test criterion.

Conclusion

The ADK_REPEAT program should be implemented as soon as possible, preferably using data from the automatic weather station no. 80700 Glomfjord for testing, as this found to contain the type of situations that the ADK_REPEAT program is specified to detect.

References

[1] P. Øgland, *AWS Quality Statistics December 1998.* KLIBAS-note no. 02/99, DNMI, Oslo, 1999.

[2] M. Kiene, C. Haeberli and R. Doessegger, Quality Control of Data from Automatic and Conventional Stations at the Swiss Meteorological Institute (SMI). *Proceedings 1st Workshop on Quality Control of Meteorologial Data in MAP,* Vienna, January 26-27, 1998.

[3] P.O. Kjensli, P. Øgland and S.L. Lystad, *Kontroll av data - spesifikasjonsrapport.* KLIBAS-report no. 01/93, DNMI, Oslo, 1993.

[4] S.L. Lystad and P. Øgland, *ADK/ADL: Datakontroll for EDAS-automatstasjoner.* KLIBAS-report no. 36/94, Oslo, 1994.

[5] P. Øgland, *Implementasjon av datakontroll ADK på Typhoon.* KLIBAS-report no. 35/94, DNMI, Oslo, 1994.

[6] P. Øgland, *Automatisk datakontroll for SYNOP og AUTO: ADK v.2.0.* KLIBAS-report no. 12/95, DNMI, Oslo, 1995.

[7] P. Øgland, *Innlesing AUTO_INN v.1.1 for AVS: Programmer mnd2ala, mnd2hla, ala2tele, adk og mkk.* KLIBAS-report no. 59/97, DNMI, Oslo, 1997.

[8] P. Øgland, *AWS real-time quality control: ADK v.2.2.* KLIBAS-report no. 55/98, DNMI, Oslo, 1998.

[9] P. Øgland, *Monthly AWS Quality Control Statistics MKK v.2.0.* KLIBAS-report no. 49/98, DNMI, Oslo, 1998.

[10] Ø. Nordli and P. Øgland, *Comparison results for the new Glomfjord automatic station versus the present manually run station.* KLIMA-report no. 04/99, DNMI, Oslo, 1999.

Handling an emergency crisis for the SYNO_INN data collection system at DNMI

Petter Øgland

Norwegian Meteorological Institute
January 18th, 1999

ABSTRACT

Whenever there are problems with datatables TELE, SYNOP or SYNOP2 growing full so that extents may not be updated for the tables, the SYNO_INN data collection system will break down. This note contains a description of how the system breaks down in one particular instance, with suggestions on how to improve in order to make it more safe.

SYNO_INN data collection

The purpose of the SYNO_INN data collection system is to read binary GTS files that are updated every 5 minutes into an Oracle RDBS as often as necessary, at the moment every 10 minutes. The present SYNO_INN program has been developed through several versions.

Using data from realtime synop data flow for insert in the climatological data base was discussed in 1985. Andresen et al. [1] wrote at the time a report addressing quality control issues in relation to observations being digitalised outside the Climatology Division.

The dataflow then implemented consisted on only reading observations relevant for long time storage in the Climatological database system on the ND computers. The dataflow was summarized in further detail during the autumn of 1993 by Øgland, Aasen and Vidal [2] as a part of the plans for the implementation of a new data collection system to be build for an Oracle RDMS on a SGI platform.

The first version of the SYNO_INN program, reading all observations from zone one on the synoXX-files into Oracle database tables, was implemented and put in operation in February 1994 (Øgland [3]) and revised due to upgrading the Oracle database system from version 6.0 to 7.0 in November 1994 [4], and revised once again by the end of the year [5] with a final revision 2.0 in April 1995 [6].

Runtime statistics from the system were systematically collected and summarised under the name GTS2FIFO in monthly reports from June 1995 and onwards (Øgland [7]). The name GTS2FIFO was chosen due to a FIFO-construction modelled after the work by Schøyen [8] on administrative data.

In October 1996 the SYNO_INN program was completely rewritten [9], later versions up to the present [10-15] being grown out of this code.

The first log of the program is in May 1997. It was then run 212 times, failing 28 of these (13.2%). From the next month and onwards, the system stabilised and recorded only fragments of problems until a new design v.3.3 in October 1998 [12]. The average number of failures then increased to about one percent.

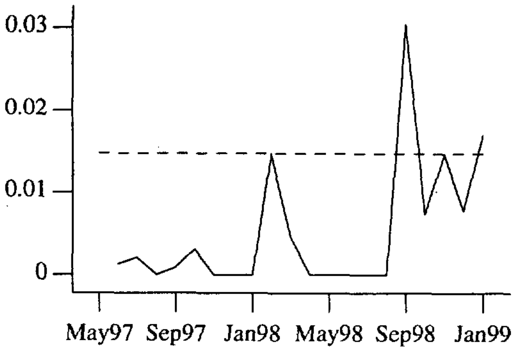


Fig 1. Relative number of failures

The average value dashed in the figure is 0.0147.

The extent problem

On Sunday the 17th of January 1999 at 19:41:52 GMT, the system once again broke down, this time due to "ORA-01654: unable to extend index V.SYNOP_PK by 6420 in tablespace VINDEX". The error was recorded in the function *insert_synop* which was called from the function *data*, called from the fucntion *system_test*, called from the function *main*.

The consequences of this fatal update of the SYNOP table thus made the complete program break down. Even if the person responsible for the SYNOP datatable had been available and could have done something to make the program be able to increase the extents, the present version of the program would have been out of function for some 16 hrs at least.

The SYNO_INN data collection is responsible for storing data that are vital to several central statistics at DNMI, not at least the daily statistics, and it seems rather dangerous that a program such as this depends on instant manual ineraction when it breaks down. What would happen if the program had broken down on a Friday afternoon? Then it would quite likely have been out of function during the weekend, and even more serious breakdown could happen if the system had broken down during the national hol idays.

Even if it is not possible to update the SYNOP data table, it does not necessarily mean that it is impossible to update the SYNOP2 and TELE data tables. For all practical purposes, the SYNOP2 and in particular the TELE data tables are the ones that are vital for producing statistics.

Suggestions for improvement

In order to avoid the complete system breaking down if the Oracle interface calls to SYNOP reports error in the function *insert_synop*, the *data* function should be reprogrammed to register the error but only act as if it was a warning. Similar reprogramming should also be done for the *insert_synop2* and *insert_tele* functions.

Even though the ERROR is not acted upon as an ordinary system error, immidiate action in order to fix the system should be taken. The error analysis in the SYNO_INN script should be augmented so that it generates automatic e-mail warnings sent to the reponsible persons so that the problem may be corrected as soon as possible.

Statistics of the type represented in figure 1 may change somewhat after these alterations as some of these severe errors are no longer treated as system errors. The errors will occur on other errors statistics, however, and will give notion both on when to act and a historical profile of how the program is failing.

References

[1] Andresen, L. et al. *Rapport nr. 1 fra datakontroll-gruppen.* KLIMA work note no. 41, DNMI, Oslo, 1985.

[2] Øgland, P., Aasen, T., Vidal, Å.M. *Data inn - spesifikasjonsrapport.* KLIBASI-report no. 01/94, DNMI, Oslo, 1994.

[3] Øgland, P. *Innlasting av synoptiske data til arbeidslager.* KLIBAS-report no. 05/94, DNMI, Oslo, 1994.

[4] Øgland, P. *Omlegging av databaserutiner ved overgang fra Oracle6 til Oracle7.* KLIBAS-report no. 28/94, DNMI, Oslo, 1994.

[5] Øgland, P. *Innlasting av synoptiske data til arbeidslager. Revidert utgave.* KLIBAS-report no. 40/94, DNMI, Oslo, 1994.

[6] Øgland, P. *Innlasting av synoptiske data til arbeidslager. Versjon 2.0.* KLIBAS-report no. 18/95, DNMI, Oslo, 1995.

[7] Øgland, P. *Driftsrapport juni 1995.* KLIBAS-note no. 09/95, DNMI, Oslo, 1994.

[8] Schøyen, A. *Brukerveiledning LOG-DATABASE V2.0.* DNMI, EDB-avdelingen, Oslo, 1994.

[9] Øgland, P. *Eksperimentell overføring av data fra syno-filer til Oracle-database. Versjon 3.0.* KLIBAS-report no. 15/96, DNMI, Oslo, 1996.

[10] Øgland, P. *Overføring av data fra syno-filer til tabeller SYNOP og TELE. Versjon 3.1.* KLIBAS-report no. 04/97, DNMI, Oslo, 1997.

[11] Øgland, P. *Dataoverføring SYNO_INN v.3.2 fra syno-filer til tabeller SYNOP og TELE med utvidet sikkerhet.* KLIBAS-report no. 55/97, DNMI, Oslo, 1997.

[12] Øgland, P. *SYNO_INN v.3.3: Revised for inserting international synops into TELE.* KLIBAS-report no. 03/98, DNMI, Oslo, 1998.

[13] Øgland, P. *Reading data from syno-files into KLIBAS: SYNO_INN v.3.4.* KLIBAS-report no. 54/98, DNMI, Oslo, 1998.

[14] Øgland, P. *Reading data from syno-files into KLIBAS: SYNO_INN v.3.5.* KLIBAS-report no. 59/98, DNMI, Oslo, 1998.

[15] Øgland, P. *Reading data from syno-files into KLIBAS: SYNO_INN v.3.6.* KLIBAS-report no. 65/98, DNMI, Oslo, 1998.

Designing procedures for adding new AERO stations to the AERO stations routine (VIND_REG) at DNMI

Petter Øgland

Norwegian Meteorological Institute
January 19th, 1999

ABSTRACT

The VIND_REG data transport program is responsible for reading automatic wind registrations from data files and inserting these into the XVIND Oracle data table in the KLIBAS database system. The system is becoming increasingly robust, but when files containing previously undefined stations, it breaks down. This note gives an analysis of one such case and some suggestions on how to solve the problem.

The VIND_REG program

Specifications for the VIND_REG program were written by Lars Andresen in September 1997, resulting in a first version of VIND_REG being released the 12th of July 1998 [1]. The purpose of the program was to read files, interpret and modify observations when necessary and store the results in suitable datatables in the KLIBAS database.

By August 18th 1998 a routine was being established at the Climatology Division for handling the observations. A quality control program VINDDEK was then realeased according to specifications by Lars Andresen and P.O. Kjensli [2].

By the end of August the same year, a revised version of VIND_REG appeared [3]. Minor faults in the algorithm had been corrected, the specification had been slightly updated.

Still there were problems with the program, and on the 20th of October 1998 a version 1.2 of VIND_REG was released [4]. The program was now running systemaically every hour of the day by the crontab schedule on the UNIX machine. A new table VINDX was designed for storing the observations.

The first log of the program is in April 1998. It was then run 17 times, failing 0 of these. The curve below shows the relative number of runs of the program that has been failing according to the log.

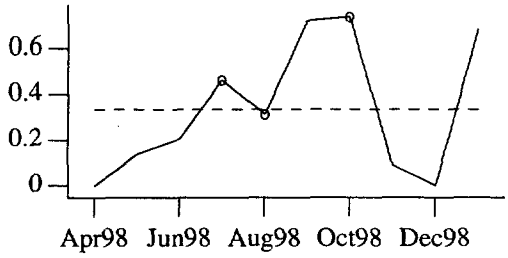


Fig 1. Relative number of failures

The average value dashed in the figure no 1. is 0.33. The time when the revisions of the program appeared are marked with a small circle.

Adding a station at Vardø to the system

On 14 January 1999 at 19:44 and 19:46 UTC, two files containing data from the station no. 98580 Vardø, identified by the code ENES, were added to the system. The files enes1198.txt and enes1298.txt contained data for November and December 1998.

The code ENES was, however, not recognised by the VIND_REG system, and was not found in digitalised form in the KLIBAS database system. If, on the other hand, such information had been available, the VIND_REG program could have read this table, found the correct identification, and there would have been no problem.

Conclusions

The VIND_PARA table used for local identificaton by the VIND_REG program should be manually updated.

A request should be made for adding information about the EN-codes in the KLIBAS database system, and if this is accomodated, the VIND_REG program should be redesigned in order to identify stations by this and not by the VIND_PARA table.

References

[1] P. Øgland, *Computer program VIND_REG.* KLIBAS-report no. 22/98, DNMI, Oslo, 1998.

[2] P. Øgland, *Computer program VINDDEK.* KLIBAS-report no. 39/98, DNMI, Oslo, 1998.

[3] P. Øgland, *Computer program VIND_REG v.1.1.* KLIBAS-report no. 43/98, DNMI, Oslo, 1998.

[4] P. Øgland, *Storing AWS wind observations by use of VIND_REG v.1.2.* KLIBAS-report no. 56/98, DNMI, Oslo, 1998.

A problem with non-defined weather elements in the KLIBAS climatological database table TELE containing unwanted data

Petter Øgland

Norwegian Meteorological Institute

January 20th, 1999

ABSTRACT

The TELE data table is used for several climatological purposes. If it should contain obviously false data, for example precipitation values for stations that do not observe precpiation, then the data table can easily lead to wrong conclusions. The present status for the TELE data table is analysed, and suggestions for improving the situation are given.

The SYNO_KONTR program

The first version of SYNO_KONTR was written as a part of the version 3.0 of SYNO_INN in October 1996 [1]. The purpose of this first program was to compare observations in the datatables SYNOP and SYNOP2 that were both updated by SYNO_INN v.3.0.

In August 1997 the revised verson 2.0 of SYNO_KONTR was released [2]. Now the program was responsible for running several quality control and interpolation programs on a daily basis and generate statistics from the execution of these programs.

The first log of the program is in July 1997. It was then run 15 times, failing 6 of these (40.0%). The curve below shows the relative number of runs of the program that has been failing according to the log.

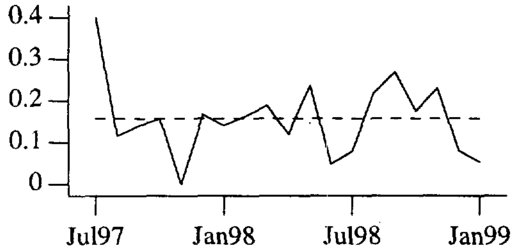


Fig 1. Relative number of failures

The average value dashed in the figure no 1. is 0.16. There is no apparent visual indication in the figure that might indicate that the program is stabilising although the two last months show a defect ratio significantly below average.

Even though the interpolation and quality control programs run by SYNO_KONTR may be perfectly under control, this is no guarantee that the data quality of the TELE data table is satisfiable. Measuring the data quality in the TELE table, on the other hand, could be an indicator of how well the SYNO_KONTR subprograms are performing.

Non-defined weather observations in TELE

By manual inspection, it has been found that there may be non-defined weather observations in TELE, meaning there may sometimes be found precipitation values for maritime weather stations that do not measure precipitation or cloud cover values for automatic weather stations that have no means for observing cloud cover.

In fact, according to definitions in TELE_PARA, there are presently 541963 undefined values in TELE among the elements TT, TN, TX, UU, RR, SS, FG, FX, P, P0, PP, A and N. Some of these may be due to TELE_PARA not containing the proper definitions, others may be due to ill defined interpolations by the INTERPOL2 [3] and INTERPOL_P0 [4] programs or wrong inserts into TELE by the SYNO_INN [5] program.

There are also observations missing in the TELE table even after the interpolation has been done. Presenent count gives 27032 missing values among the mentioned weather elements.

Planning a study of the content of TELE

Before further improvements on, say, the INTERPOL2 interpolation program, investigating why interpolation is sometimes done when it should not, the TELE_PARA table have to be updated so that non-defined observations according to TELE_PARA do, in fact, correspond to observations that are really non-defined.

A natural place to automatically update the TELE_PARA table, would perhaps be the SYNO_INN program, which presently is used for automatic updates in TELE_PARA of which hours of the day each station is expected to report data.

Before augmenting the SYNO_INN program, however, SYNO_KONTR should be made to count the daily number of non-defined and missing observations in the TELE table according to TELE_PARA definitions, so that it will be possible to investigate the gradual quality improvement of the TELE table, and, more importantly, be able to detect tendencys that may imply decrease in quality.

References

[1] P. Øgland, *Eksperimentell overføring av data fra syno-filer til Oracle-database. Versjon 3.0.* KLIBAS-report no. 15/96, DNMI, Oslo, 1996.

[2] P. Øgland, *Døgnlig driftskontroll for SYNOP: syno_kontr v.2.0.* KLIBAS-report no. 60/97, DNMI, Oslo, 1997.

[3] P. Øgland, *Quality control computer program S-T-F v.2.1: Automatic removal of "bad" INTERPOL2 values.* KLIBAS-report no. 17/98, DNMI, Oslo, 1998.

[4] P. Øgland, *Interpolating air pressure in TELE with INTERPOL_P0 v.1.1.* KLIBAS-report no. 52/98, DNMI, Oslo, 1998.

[5] P. Øgland, *Reading data from syno-files into KLIBAS: SYNO_INN v.3.7.* KLIBAS-report no. 05/99, DNMI, Oslo, 1999.

A problem of non-defined weather elements being interpolated in the KLIBAS climatological database table TELE

Petter Øgland

Norwegian Meteorological Institute
January 25th, 1999

ABSTRACT

The programs INTERPOL2 and INTERPOL_P0 are reponsible for updating the TELE datatable with estimates of meteorological values, such as temperature, TT, TN, TX, relative humidity UU, preciptation RR, air pressure P, P0, A, PP and cloud cover N. By inspection it has been registered, however, that in some instances values may be estimated and inserted into TELE although the particular station in question does not measure the suggested weather element. In this paper such a case is described, analysed and su ggestions on how to improve on the situation are given.

The TELE interpolation programs

The first attempts at automatic interpolation in TELE were done in June 1994 [1], but this was only a test and the code for this particular program was not develloped into full maturity.

The first version of the present interpolation system, INTERPOL2, was released in August 1997 [2]. The interpolation technique was evaluated [3] and then improved [4].

In May 1998 the interpolation program was made to communicate with the quality control program S-T-F by having S-T-F remove observations inserted into TELE by INTERPOL2 when they were in conflict with the S-T-F quality control [5].

In order to improve further on the interpolation system, a special module for interpolating air pressure, INTERPOL_P0, was constructed in September 1998 [6]. This program is running in sequence with INTERPOL2 in order to overwrite any air pressure interpolations done by INTERPOL2 and to compare the quality of the estimates done by the two programs.

A revised version of INTERPOL_P0 was released by the end of September 1998, using an improved strategy for interpolating air pressure [7].

All interpolation and quality control programs are run systematically twice a day from a program SYNO_KONTR which is also responsible for checking the content of the TELE data table after all updates have been done [8].

The first log of the INTERPOL2 program is in August 1997. It was then run 92 times, failing 8 of these (8.7%). The curve below shows the relative number of runs of the program that has been failing according to the log.

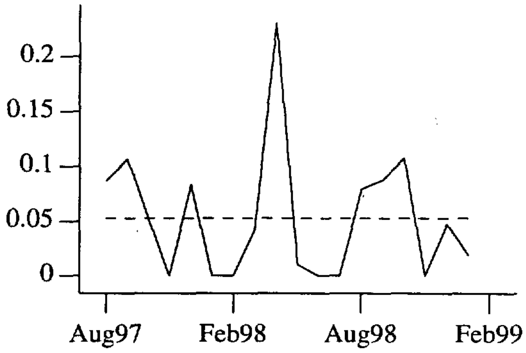


Fig 1. Relative number of failures

The average value dashed in the figure no 1. is 0.05. None of these problems have anything to do with updating the wrong columns.

Cloud cover at Røst III

As an example of the problem with INTERPOL2 updating columns in TELE that are not defined according to TELE_PARA, we may regard the data set from the automatic weather station (AWS) no. 85891 at Røst III. As this AWS does not record cloud cover (N), it does not make sense for the interpolation programs to estimate N and fill in these values in the TELE table.

Looking at the data in TELE, there are interpolated values of N from November 1998 and onwards, although some have been deleted manually, which can be seen by looking at the FLN flag that have been updated from '4' to '1'. Interpolated values may also have been updated manually without updating the FLN flag column.

Investigating the TELE table for the station no. 85891, it becomes clear that the FLN flag is either '1', '4' or does have no value at all for this station. In every case where FLN is '1', N is empty. This means that all cloud cover information for this station has been estimated and inserted by the INTERPOL2 program. The column N in the TELE_PARA table, corresponding to station no. 85891 contains no value.

Why does INTERPOL2 insert cloud cover?

The intended logic of the INTERPOL2 program was that only updates in TELE should be done when the result of comparing observations at the station with neighbouring stations would indicate that the estimates were sufficiently good not to produce gross error ([2], page 6).

As the analysis of Røst III shows, this is not how the program works in practice. Contrary to intensions, values are inserted when there is absolutely no guarantee that they represent the weather at the station. In fact, as there are no cloud cover observations at Røst III, even inserting random values as estimates would technically not prove any worse than the present method as there are no reference data to compare with.

One defect that may be partial cause to the problem is that the MANGLLISTE program used by INTERPOL2 to pinpoint missing values does not seems to read the TELE_PARA properly when describing missing values. In the same manner as INTERPOL2 it may report a missing value from a particular weather element when the station does not observe this weather element.

The problem with the MANGELLISTE program does not, however, fully explain why the INTERPOL2 program interpolates weather elements that are not being recorded by the station. An internal check structure in the INTERPOL2 program was originally designed in order to prohibit it from making updates of this kind.

There are three ways the INTERPOL2 program may prohibit estimates being used for interpolation:

1. In the function "file2ref" there is a check which reports "no interpolation due to non-accessible raw data" if the test station cannot be found on the data file. By checking the err files, it appears that a warning from this test has not been generated since 24th September 1998.
2. Average error and standard deviation of error are computed and compared with limits that are given by the shell program. If these limits are exceeded, all estimates will be removed and no updated done. None of these warnings have been found on files, however, perhaps as a consequence of the limits being put so high that they in practice never will be exceeded.
3. The third instance where the estimates may be removed are in the "ref2file" function when the sum of weights from the ST_REF molecule (struct) amounts to zero, meaning that the estimate is zero due to no estimation being done as the estimator would automatically be set like zero. This case will, however, not produce a warning.

The conclusion seems to be that there is really nothing that prevents the columns to be updated if estimates have been made. An RMSE evaluation is done for each element, but this is done independently of whether the columns are being udpated or not.

Conclusion

First the INTERPOL2 program should be altered so that all updates in TELE are done in correspondence with the definitions in TELE_PARA. Then the TELE data table should be updated, removing all interpolated values that are not defined in TELE_PARA.

Also the MANGELLISTE program should be corrected, so only those values corresponding to weather elements that are acutally under observation at a particular site are being reported missing.

References

[1] P. Øgland, *Eksperimentering med enkel kvalitetskontroll og interpolasjon av*

værstasjonsdata med varierende tidsoppløsning. KLIBAS-report no. 18/94, DNMI, Oslo, 1994.

[2] P. Øgland, *Automatic interpolation of SYNOP weather observations: Interpoll and Interpol2.* KLIBAS-report no. 61/97, DNMI, Oslo, 1997.

[3] P. Øgland, *Evaluation of a meteorological radial interpolation method by statistical experiments.* KLIMA-report no. 23/97, DNMI, Oslo, 1997.

[4] P. Øgland, *Evaluation of a double exponential correlation weighted interpolation method by statistical experiments.* KLIMA-report no. 25/97, DNMI, Oslo, 1997.

[5] P. Øgland, *Quality control computer program S-T-F v.2.1: Automatic removal of "bad" INTERPOL2 values.* KLIBAS-report no. 17/98, DNMI, Oslo, 1998.

[6] P. Øgland and Z. Toporowski, *Interpolating air pressure in TELE with INTERPOL_P0.* KLIBAS-report no. 46/98, DNMI, Oslo, 1998.

[7] P. Øgland and Z. Toporowski, *Interpolating air pressure in TELE with INTERPOL_P0 v.1.1.* KLIBAS-report no. 52/98, DNMI, Oslo, 1998.

[8] P. Øgland, *SYNO_KONTR v.2.1: Daily interpolation and quality control.* KLIBAS-report no. 06/99, DNMI, Oslo, 1999.

Solving a segmentation fault problem in the computer assisted computer programming (CACP) KLIBAS routine at DNMI

Petter Øgland

Norwegian Meteorological Institute

January 28th, 1999

ABSTRACT

The CACP-routine KAPO in the KLIBAS system at DNMI is used daily for updating log files and keeping the software development under strict control. A collapse of this system inflict total collapse in software development. In this paper one such collapse is analysed with suggestions on how to make the program operative again.

The KAPO program

Software development by use of standards was a major concern in 1992. A report containing statndards for systems development at the Climatology Division was written with emphasis on programming, user interfaces, computer interfaces and documentation among other things [1].

At the same time, effort was put into how projects should be run by emphasis on quality management [2]. Special concern was put on the NS-ISO 9000 series, use of a quality handbook and internal quality revisions. In November 1994 an internal quality revision system was implemented [3], and a quality handbook for the KLIBAS project was written [4] which was enlarged by a second volume in January 1995 [5].

In 1997 a series of efforts was made of collecting system information from the KLIBAS database system in order to reduce risico factors for the system breaking down and getting more control over how resources was being spent on system development [6]. Special concern was put on finding trends in measured quality [7] and ways of handling warning messages automatically [8].

Data quality was also a part of the general system monitoring [9], and so was monitoring of the performance of the database server and the Oracle database core [10]. As system email messages became an increasing part of the system communications, programs for checking and handling these were also developed [11].

In Mars 1998 automatic analysis of the crontab scheduling in order to control programs that were running at times when the Oracle database was not available was added [12]. Automatic changes in the crontab scheduling when conflicts rose was also added [13], and soon a system for deleting old files or dangerously large files was a part of the general KLIBAS system [14].

In September 1998 further thoughts on bringing the KLIBAS database system and work performed in the KLIBAS environemnt within the restrictions of ISO 9000 was planned [15]. The computer program KAPO for computer assisted computer programming was one such vital ingrediance [16]

The first log of the KAPO program is traced back to December 1997. It was then run 17 times, failing 4 of these (23.5%). The number of users during this first month was 1. The curve below shows the relative number of runs of the program that has been failing according to the log.

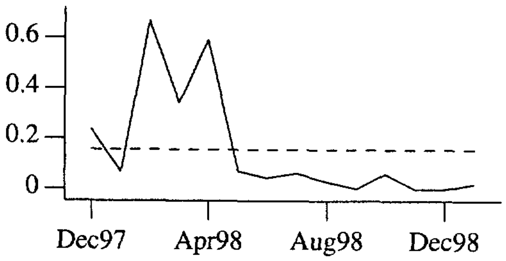


Fig 1. Relative number of failures

The average value dashed in the figure is 0.17, although non of the recording after the significant improvement on quality in Mars 1998 show a rate of more than 0.07 abnormal ternimations pr run.

The segmentation fault in January 1999

When trying to run the KAPO program on the 28th of January 1999, it collapsed in uncontrolled mode, leaving the error message "segmentation fault".

As the breakdown is uncontrolled, finding the source for the problem has to be done by testing and careful analysis. The symptoms file for the program only show that it has reached the point where it is about to process data as the message "134 programs have been read from priplan.inp file" is the last message displayed.

By systematic testing it was found, however, that the reason for failure had to do with how the KAPO program reads the list of problems from the output file generated by the DRIFT program. The DRIFT program had been altered on the 27th of January.

The KAPO program reads the DRIFT result file line by line and identifies each lines written in bold face containing one of the words "warning" or "error" as a line containing information on the number of warnings or errors for a particular program.

In order to solve this particular problem, KAPO should be rewritten to contain a more sophisticated method of identifying this particular lines in the DRIFT output. More specifically, using for instance brackets as symbols in the part of the text to be regonised should solve the problem.

More important, however, is the question on how to prevent problems is this kind arising in the future. One possibility could, of course, be to program an even more sophisticated pattern recognition algoritm, but as the combination of bold face, brackets and one of the words "warning" or "error" to be present in the wrong type of line, seems rather unlikely, the present suggestions for adjustment of the code may be sufficient.

References

[1] M. Moe et al, *Standarder for systemutvikling. Delprosjekt 4.* KLIMA-report no. 44/92, DNMI, Oslo, 1992.

[2] P.O. Kjensli et al., *Kvalitetsstyring for prosjektarbeid. Delprosjekt 5.* KLIMA-report no. 45/92, DNMI, Oslo, 1992.

[3] P. Øgland et al., *Praktisk rutine for kvalitetssikring av programvare.* KLIBAS-report no. 33/94, DNMI, Oslo, 1994.

[4] P. Øgland et al., *Kvalitetshåndbok for databaseprosjektet i Klimaavdelingen.* KLIBAS-report no. 34/94, DNMI, Oslo, 1994.

[5] P. Øgland et al., *Kvalitetshåndbok for databaseprosjektet i Klimaavdelingen. Del II.* KLIBAS-report no. 03/95, DNMI, Oslo, 1995.

[6] P. Øgland, *Logging av driftsparametre for kvalitetssikring av driftsrutiner.* KLIBAS-report no. 45/97, DNMI, Oslo, 1997.

[7] P. Øgland, *Kvalitetstrend QUAL_TREND i månedlig driftsrapportering.* KLIBAS-report no. 46/97, DNMI, Oslo, 1997.

[8] P. Øgland, *Eksperiment med automatisk feilbehandling i KLIBAS.* KLIBAS-report no. 49/97, DNMI, Oslo, 1997.

[9] P. Øgland, *Beregning og presentasjon av datakvalitet i månedlig driftsrapport.* KLIBAS-report no. 50/97, DNMI, Oslo, 1997.

[10] P. Øgland, *Beregning og presentasjon av maskinvarekvalitet i månedlig driftsrapport.* KLIBAS-report no. 53/97, DNMI, Oslo, 1997.

[11] P. Øgland, *Computer program CHECK_MAIL.* KLIBAS-report no. 06/98, DNMI, Oslo, 1998.

[12] P. Øgland, *Analysing CRONTAB scheduling with computer program ORACLE_SHUTDOWN.* KLIBAS-report no. 13/98, DNMI, Oslo, 1998.

[13] P. Øgland, *Automatically altering CRONTAB with computer program CHECK_15MIN v.2.0.* KLIBAS-report no. 14/98, DNMI, Oslo, 1998.

[14] P. Øgland, *Computer program GRIM_REAPER.* KLIBAS-report no. 15/98, DNMI, Oslo, 1998.

[15] P. Øgland, *Towards an ISO-9000 standard for system development by DRIFT.* KLIBAS-report no. 47/98, DNMI, Oslo, 1998.

[16] P. Øgland, *Computer assisted computer programming: KAPO v.1.0.* KLIBAS-report no. 02/99, DNMI, Oslo, 1999.