



DNMI

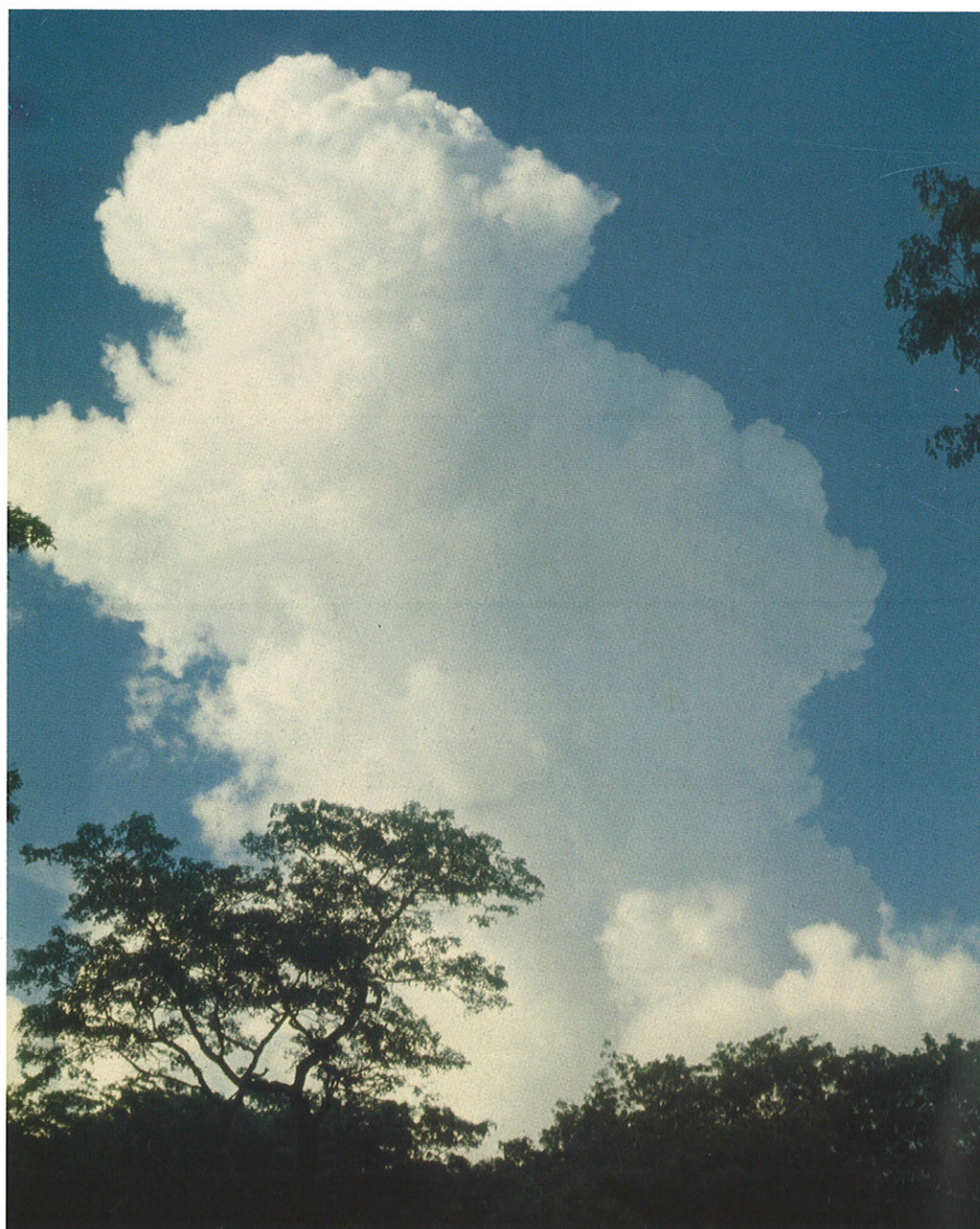
Det norske meteorologiske institutt

Report no. 15/99

KLIMA

KLIBAS research notes volume 5

Petter Øgland (ed.)



DNMI - REPORT

ISSN 0805-9918

NORWEGIAN METEOROLOGICAL INSTITUTE
BOX 43 BLINDERN N-0313 OSLO

REPORT NO.

15/99 KLIMA

DATE

May 1 1999

PHONE: +47 22 96 30 00

TITLE

KLIBAS RESEARCH NOTES VOLUME 5

AUTHOR

Petter Øgland (ed.)

PROJECT CONTRACTOR

DNMI - Climatology Division / NORDKLIM

SUMMARY

In this volume 15 research notes dating from April 1999 are presented. All notes are related to on-going research and development of the KLIBAS climatological database system.


The problems analysed split into six categories.

1. Automatic weather stations (AWS).
2. Automatic climate stations (AANDERAA).
3. The XVIND automatic weather stations (VIND_REG).
4. The TELE routine.
5. The PIO data processing routine.
6. General KLIBAS system improvements.

KEYWORDS

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

SIGNATURE


.....
Petter Øgland

Research Scientist


.....
Bjørn Aune

Head of Climatology Division

Table of Contents

Foreword	3
Note no. 1: "Making the computers Gale and Halo communicate and control each other by mail checking"	5
Note no. 2: "Using SPC for NORDKLIM/KLIBAS project management"	8
Note no. 3: "Adapting VSUKE for Quality Control of TELE observations"	11
Note no. 4: "Forcing old tasks on the WBS list in the DRIFT program to be managed by the KLIBAS database system"	14
Note no. 5: "Automatically updating the KLIBAS web page"	17
Note no. 6: "Improving on a temporary solution for the VIND_REG program in the the XVIND routine at DNMI"	19
Note no. 7: "Daily quality evaluation for automatic weather stations by AUTO_MKK v.1.1"	22
Note no. 8: "Producing statistics for daily monitoring of the AANDERAA routine"	25
Note no. 9: "Adding a control feature in CHECK_MAIL in order to make sure that the AUTO_MKK program is running"	28
Note no. 10: "Adding features in MAILSTAT in order to remove warning messages and check up on the CHECK_MAIL program"	31
Note no. 11: "Adding linear interpolation of air temperature to the ALA2TELE data transfer program"	34
Note no. 12: "Improving upon a solution for the PIO_INN data collection system"	36
Note no. 13: "Eliminating an ORACLE ERROR SQL-02112: "SELECT..INTO returns too many rows" from the SYNO_INN computer program"	38
Note no. 14: "Preventing the SYNO_KONTR system from updating temperature where FLTT equals one"	41
Note no. 15: "Solving a problem in the ALA2TELE_PARA program"	45

Foreword

This 5th volume of KLIBAS research notes contains a documentation of problems studied in April 1999. While most of the problems addressed in this collection of notes have later been solved [1-19], 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. Six areas of climatological database research have been targeted for problem analysis.

1. *The XVIND automatic weather stations (VIND_REG)*. Check note no. 6 "Improving on a temporary solution for the VIND_REG program in the the XVIND routine at DNMI".
2. *The TELE data processing routine*. Check note no. 3 "Adapting VSUKE for Quality Control of TELE observations". Check note no. 11 "Adding linear interpolation of air temperature to the ALA2TELE data transfer program". Check note no. 13 "Eliminating an ORACLE ERROR SQL-02112: "SELECT..INTO returns too many rows" from the SYNO_INN computer program". Check note no. 14 "Preventing the SYNO_KONTR system from updating temperature where FLTT equals one". Check note no. 15 "Solving a problem in the ALA2TELE_PARA program".
3. *The PIO data processing routine*. Check note no. 12 "Improving upon a solution for the PIO_INN data collection system".
4. *The automatic weather stations (AWS)*. Check note no. 7 "Daily quality evaluation for automatic weather stations by AUTO_MKK v.1.1".
5. *The automatic climate stations (AANDERAA)*. Check note no. 8 "Producing statistics for daily monitoring of the AANDERAA routine".
6. *General KLIBAS system improvement*. Check note no. 1 "Making the computers Gale and Halo communicate and control each other by mail checking". Check note no. 2 "Using SPC for NORD-KLIM/KLIBAS project management". Check note no. 4 "Forcing old tasks on the WBS list in the DRIFT program to be managed by the KLIBAS database system". Check note no. 5 "Automatically updating the KLIBAS web page". Check note no. 9 "Adding a control feature in CHECK_MAIL in order to make sure that the AUTO_MKK program is running". Check note no. 10 "Adding features in MAILSTAT in order to remove warning messages and check up on the CHECK_MAIL program".

The notes in this volume document a number of problems in the KLIBAS database system during April 1999. The reason why these particular problems were being addressed this month had to do with the automatically updated priority list presented in the status report for March 1999 [1].

Petter Øgland
Blindern, May 1st 1999

References:

- [1] DNMI/Klima, *Interpolation and Quality Control Status for March 1999*, DNMI-Note no. 07/99 KLIBAS, Oslo, April 1999.
- [2] DNMI/Klima, *KLIBAS research notes volume 4*, DNMI-Report no. 13/99 KLIMA, Oslo, April 1999.
- [3] DNMI/Klima, *Adding SPC and communication features in CHECK_MAIL v.1.1*, DNMI-Report no. 24/99 KLIBAS, Oslo, April 1999.

- [4] DNMI/Klima, *Automatic update of NORDKLIM project plans by DRIFT v.1.1*, DNMI-Report no. 25/99 KLIBAS, Oslo, April 1999.
- [5] DNMI/Klima, *Using VSUKE v.2.1 in Quality Control*, DNMI-Report no. 26/99 KLIBAS, Oslo, April 1999.
- [6] DNMI/Klima, *Automatical update of the KLIBAS web-page by WEB v.1.0*, DNMI-Report no. 27/99 KLIBAS, Oslo, April 1999.
- [7] DNMI/Klima, *Adding process control features in VIND_REG v.1.3*, DNMI-Report no. 28/99 KLIBAS, Oslo, April 1999.
- [8] DNMI/Klima, *Daily Quality Evaluation of AWS by the program AUTO_MKK v.1.1*, DNMI-Report no. 29/99 KLIBAS, Oslo, April 1999.
- [9] DNMI/Klima, *Using AANDERAA_INN v.1.1 for monitoring AWS data collection*, DNMI-Report no. 30/99 KLIBAS, Oslo, April 1999.
- [10] DNMI/Klima, *AWS Quality Evaluation Statistics: March 1999*, DNMI-Note no. 08/99 KLIBAS, Oslo, April 1999.
- [11] DNMI/Klima, *Monitoring KLIBAS system mail by MAILSTAT v.2.0*, DNMI-Report no. 31/99 KLIBAS, Oslo, April 1999.
- [12] DNMI/Klima, *Adding linear interpolation of temperature in ALA2TELE v.1.1*, DNMI-Report no. 32/99 KLIBAS, Oslo, April 1999.
- [13] DNMI/Klima, *Draft report to the first NORDKLIM QC meeting. Helsinki, April 22-23 1999*, DNMI-Report no. 33/99 KLIBAS, Oslo, April 1999.
- [14] DNMI/Klima, *PC i observasjonstjenesten: PIO_INN v.2.3*, DNMI-Report no. 34/99 KLIBAS, Oslo, April 1999.
- [15] DNMI/Klima, *Monitoring KLIBAS system mail by MAILSTAT v.2.1*, DNMI-Report no. 35/99 KLIBAS, Oslo, April 1999.
- [16] DNMI/Klima, *SYNO_KONTR v.2.2: Daily routine for quality control in TELE*, DNMI-Report no. 36/99 KLIBAS, Oslo, April 1999.
- [17] DNMI/Klima, *HIRLAM v.1.1: Interpolating in TELE with forecast data*, DNMI-Report no. 37/99 KLIBAS, Oslo, April 1999.
- [18] DNMI/Klima, *Inserting snr/stnr into TELE_PARA by ALA2TELE_PARA v.1.2*, DNMI-Report no. 38/99 KLIBAS, Oslo, April 1999.
- [19] DNMI/Klima, *Updating SAWS in ALV and TELE by PIO2ALV v.1.0*, DNMI-Report no. 39/99 KLIBAS, Oslo, April 1999.

Making the computers Gale and Halo communicate and control each other by mail checking

Petter Øgland

Norwegian Meteorological Institute
Niels Henrik Abels vei 40
P.O. Box 43, Blindern
N-0313 Oslo, NORWAY
(subm. April 2, 1999)

ABSTRACT

The program CHECK_MAIL was constructed in order to automatically delete old mails for system user kabase. If not deleted automatically, the mailbox of kabase has a tendency of growing far to full before mails are deleted manually. The problem has been noted on the monthly list of problems (as "logging av backup") since September 1995. This note addresses a problem with the program causing it to top the problem statistics in the KLIBAS maintenance system.

The CHECK_MAIL program

February 1998: The program CHECK_MAIL [1] was designed for automatically deleting redundant mail for system user KABASE.

February 1999: The first documented version of the MAILSTAT program so far only collects statistics from mailbox. Further programming will be needed for presenting the statistics as more values are collected. The program in its present form is important, however, as an aid for understanding how the KLIBAS warning mail system performs as priority lists for maintenance and development is based on this system.

April 1999: As it was discovered that the original CHECK_MAIL system [3] had not been running for several months, allowing the mailbox for KABASE to grow uncontrollably large, it was apparent that the program needed better monitoring. In the version 1.1 of CHECK_MAIL, it updates a daily SPC chart and an automatic mail communication between the users KABASE and KAPO is established.

Each run by CHECK_MAIL is logged, registering program breakdown as a defect. The first log of the program is in February 1998. It was then run 52 times, failing 10 of these (19.2%).

The number of users during this first month was 2. The curve below shows the relative number of runs of the program that has been failing according to the log.

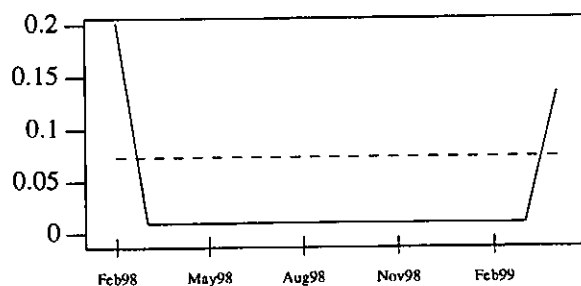


Figure 1. Relative number of defects

The average value dashed in the figure is 0.06. Statistics of this kind is present method for documenting general system quality and improvement.

In order to focus not only on long term problems caused by breakdown of the program, a circular structure of the last twelve months is used for recording errors and warnings from the program.

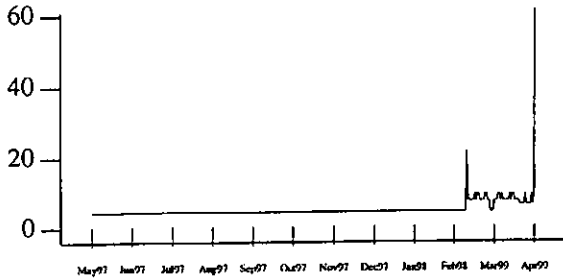


Figure 2. Problems day by day

The type of curve represented by figure 2 is used for checking whether there have been much update of the program recently or if the program is systematically reporting a great number of problems.

The current problem

The message "Error: the first problem on the mail list should be solved" was first recorded on an err file on Sat Feb 20 1999. We ask ourselves what can be learned about the KLIBAS system and CHECK_MAIL by understanding this problem, and how may it help achieving the ultimate goal of the system?

The ultimate goal of the KLIBAS system is to be totally self-contained in terms of automatically carrying out all operations of quality control of climate data with as little manual interaction as possible [4].

In order both to understand the problem and to solve it, a plan must be devised, an experiment must be carried out and the results must be interpreted.

Planning the experiment

The program CHECK_MAIL was constructed in order to automatically delete old mails for system user kabase. If not deleted automatically, the mailbox of kabase has a tendency of growing far to full before mails are deleted manually. The problem has been noted on the monthly list of problems (as "logging av backup") since September 1995.

The figure shows the day by day count of letters in the kabase mailbox on GALE. The plot is a part of the Statistical Process Control (SPC) for the system and consists of a dashed line representing the average value of the day for the process and the upper limit of the process (process capability) given in terms of six times the standard deviation of the process.

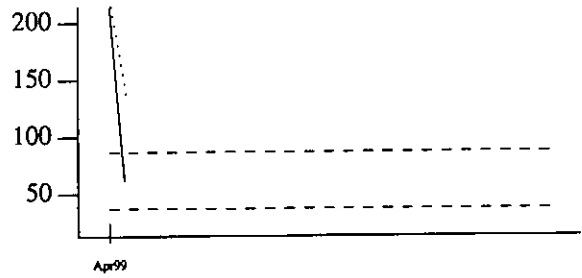


Figure 3. CHECK_MAIL evaluation by SPC

Running the experiment

According to specifications above, an experiment should be run and evaluated. The actual running of the experiment should be handled by the systematic running of the CHECK_MAIL program itself.

The CHECK_MAIL program is run automatically once a day by the user KABASE on GALE as schedules by crontab. The program is responsible for deleting excess mail in the kabase mailbox and to communicate to the user KAPO that everything is okay. The communication is done via email so if KAPO and MAILSTAT were to be located on a random computer, not connected to GALE in anyway, the users still communicated and may be able to give eachother information and instructions.

When the system has been running for sufficiently long time, it should again be evaluated, this note being a vital reference for documenting improvements.

Conclusion

As a step in solving the title problem in this note, an experiment has been designed, and measures on how to carry this experiment out have been described. The results will be evaluated when enough data is available for drawing conclusions.

However, in order to put the results in a wider context and prepare for the next set of experiments, literature in the field of research should be investigated in case there are methods and concepts that should be included before new experiments are designed.

References

- [1] DNMI-Report no. 06/98 KLIBAS "Computer program CHECK_MAIL.", Oslo, February 11th, 1998.

- [2] *DNMI-Report no. 13/99 KLIBAS "Monitoring KLIBAS system mail by MAILSTAT v.1.0."*, Oslo, February 22nd, 1999.
- [3] *DNMI-Report no. 24/99 KLIBAS "Adding SPC and communication features in CHECK_MAIL v.1.1."*, Oslo, April 2nd, 1999.
- [4] *DNMI-Report no. 12/99 KLIMA "Quality Control of Climate Data: An Automatic Approach"*, Oslo, March 19th, 1999.

Using SPC for NORDKLIM/KLIBAS project management

Petter Øgland

Norwegian Meteorological Institute
Niels Henrik Abels vei 40
P.O. Box 43, Blindern
N-0313 Oslo, NORWAY
(subm. April 3, 1999)

ABSTRACT

The purpose of the computer program DRIFT is provide statistics for helping manage computer program development of the KLIBAS database system as a Software Process Improvement task. The program collects log from the KLIBAS system and generates statistics in order to assist the improvement circle. This note addresses a problem with the program causing it to top the problem statistics in the KLIBAS maintenance system.

The DRIFT program

September 1998: The DRIFT program [1] 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.

April 1999: In the version 1.1 of DRIFT [2], both the DRIFT program and KLIBAS/NORDKLIM progress is monitored by a revised type of statistical process control (SPC), based on counting the number of tasks not yet being solved according to Work Breakdown Structure (WBS). Statistics will be used in the monthly status reports.

Each run by DRIFT is logged, registering program breakdown as a defect. The first log of the program is in December 1997. It was then run 12 times, failing 2 of these (16.7%). 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.

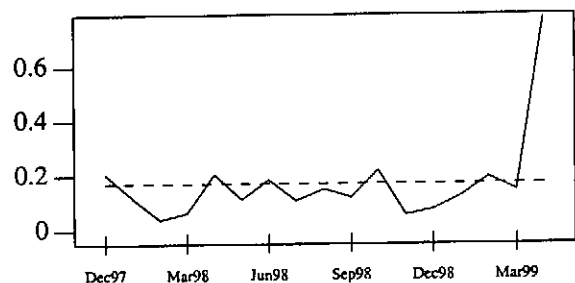


Figure 1. Relative number of defects

The average value dashed in the figure is 0.13. Statistics of this kind is present method for documenting general system quality and improvement.

In order to focus not only on long term problems caused by breakdown of the program, a circular structure of the last twelve months is used for recording errors and warnings from the program.

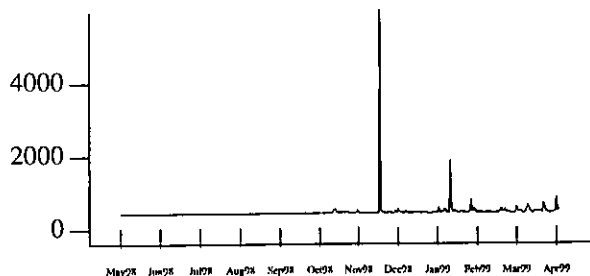


Figure 2. Problems day by day

The type of curve represented by figure 2 is used for checking whether there have been much update of the program recently or if the program is systematically reporting a great number of problems.

The current problem

The message "Error: update TELE routine diagrams in DRIFT report in order to include VSUKE and Jan-Erik in total qua" was first recorded on an err file on Thu Mar 04 1999. We ask ourselves what can be learned about the KLIBAS system and DRIFT by understanding this problem, and how may it help achieving the ultimate goal of the system?

The ultimate goal of the KLIBAS system is to be totally self-contained in terms of automatically carrying out all operations of quality control of climate data with as little manual interaction as possible [3].

In order both to understand the problem and to solve it, a plan must be devised, an experiment must be carried out and the results must be interpreted.

Planning an experiment

The purpose of the computer program DRIFT is provide statistics for helping manage computer program development of the KLIBAS database system as a Software Process Improvement task. The program collects log from the KLIBAS system and generates statistics in order to assist the improvement circle.

The figure shows the day by day count of unsolved tasks in the NORDKLIM/KLIBAS project. The WBS (Work Breakdown Structure) for KLIBAS was redefined on Sat Apr 3 1999 by coordinating it with a preliminary NORDKLIM WBS. The plot is handled as with Statistical Process Control (SPC) by also plotting dashed control lines and process capability in terms of six times the standard deviation of the process.

The most recent problem with the DRIFT system is a failure to log parts of the KLIBAS system having to do with the VSUKE program in the TELE routine. The augmentation of the system is expected to show on the SPC curve during April.

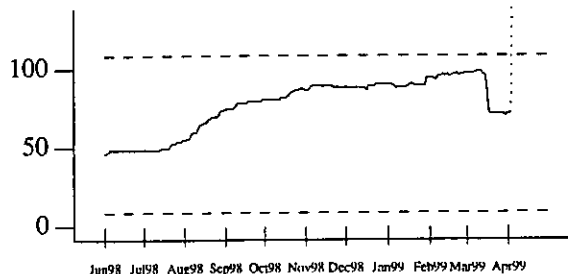


Figure 3. DRIFT evaluation by SPC

Running the experiment

According to specifications above, an experiment should be run and evaluated. The actual running of the experiment should be handled by the systematic running of the DRIFT program itself.

The DRIFT program is run on a daily basis by KAPO in order to assist in deciding which job of the day that should be carried out. It is also run automatically twice a day by crontab via KAPO.

When the system has been running for sufficiently long time, it should again be evaluated, this note being a vital reference for documenting improvements.

Conclusion

As a step in solving the title problem in this note, an experiment has been designed, and measures on how to carry this experiment out have been described. The results will be evaluated when enough data is available for drawing conclusions.

When the curve starts to drop in a systematic manner, plans should be made in order to estimate how long it will take before the curve reaches bottom and the complete WBS has been solved.

However, in order to put the results in a wider context and prepare for the next set of experiments, literature in the field of research should be investigated in case there are methods and concepts that should be included before new experiments are designed.

References

- [1] *DNMI-Report no. 47/98 KLIBAS "Towards an ISO-9000 standard for system development by DRIFT."*, Oslo, September 7th, 1998.
- [2] *DNMI-Report no. 25/99 KLIBAS "Automatic update of NORDKLIM project plans by DRIFT v.1.1."*, Oslo, April 3rd, 1999.
- [3] *DNMI-Report no. 12/99 KLIMA "Quality Control of Climate Data: An Automatic Approach"*, Oslo, March 19th, 1999.

Adapting VSUKE for Quality Control of TELE observations

Petter Øgland

Norwegian Meteorological Institute
Niels Henrik Abels vei 40
P.O. Box 43, Blindern
N-0313 Oslo, NORWAY
(subm. April 5, 1999)

ABSTRACT

The purpose of the program VSUKE is to construct standard weekly weather statistics that is needed by regular customers. This note addresses a problem with the program causing it to top the problem statistics in the KLIBAS maintenance system.

The VSUKE program

March 1994: Programs VSUKE, NSUKE and VSDUMP [1], have been designed for producing weekly weather statistics based on datatables containing up to date weather observations and daily precipitation observations in the KLIBAS database system.

January 1995: The report [2] contains updated documentation for version 2.1 of RRUTM and version 1.1 of VSUKE.

March 1997: The revision of VSUKE [3] made it possible to generate sequences of statistical output.

April 1999: The version 2.1 of VSUKE [4] contains log files for weather statistics that the program has not been able to produce. The redesign of the program has been done in order to guarantee a systematic quality check of statistics produced for regular DNMI customers.

Each run by VSUKE is logged, registering program breakdown as a defect. The first log of the program is in June 1996. It was then run 330 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.

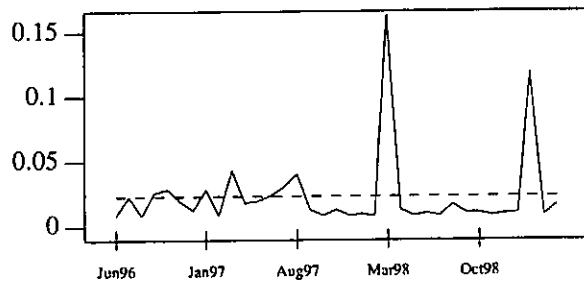


Figure 1. Relative number of defects

The average value dashed in the figure is 0.01. Statistics of this kind is present method for documenting general system quality and improvement.

In order to focus not only on long term problems caused by breakdown of the program, a circular structure of the last twelve months is used for recording errors and warnings from the program.

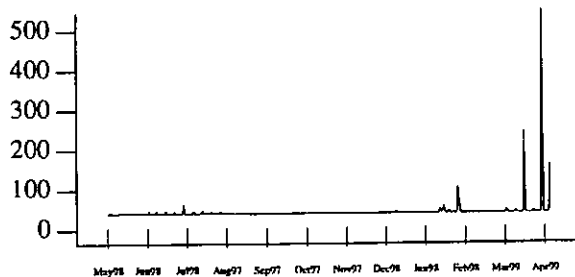


Figure 2. Problems day by day

The type of curve represented by figure 2 is used for checking whether there have been much update of the program recently or if the program is systematically reporting a great number of problems.

The current problem

The message "Error: update TELE routine diagrams in DRIFT report in order to include VSUKE and Jan-Erik in total qua" was first recorded on an err file on Thu Mar 04 1999. We ask ourselves what can be learned about the KLIBAS system and VSUKE by understanding this problem, and how may it help achieving the ultimate goal of the system?

The ultimate goal of the KLIBAS system is to be totally self-contained in terms of automatically carrying out all operations of quality control of climate data with as little manual interaction as possible [5].

In order both to understand the problem and to solve it, a plan must be devised, an experiment must be carried out and the results must be interpreted.

Planning an experiment

The purpose of the program VSUKE is to construct standard weekly weather statistics that is needed by regular customers.

The figure shows the day by day count of defects in the VSUKE weather statistics, using indicators of uncalculated values as definition of defect. The plot is a part of the Statistical Process Control (SPC) for the system and consists of a dashed line representing the average value of the day for the process and the upper limit of the process (process capability) given in terms of six times the standard deviation of the process.

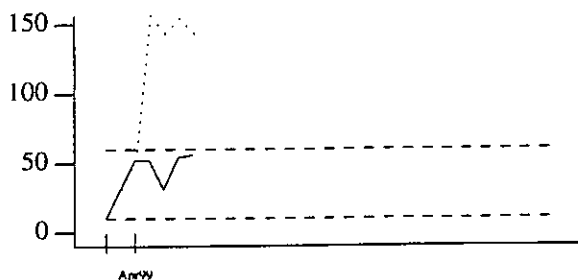


Figure 3. VSUKE evaluation by SPC

Running the experiment

According to specifications above, an experiment should be run and evaluated. The actual running of the experiment should be handled by the systematic running of the VSUKE program itself.

The VSUKE program is not run by crontab, but only executed from the synop menu a couple of times a week, mostly on Mondays as this is when regular customers expect the statistics. Often the program is run in sequence for a number of stations.

When the system has been running for sufficiently long time, it should again be evaluated, this note being a vital reference for documenting improvements.

Conclusion

As a step in solving the title problem in this note, an experiment has been designed, and measures on how to carry this experiment out have been described. The results will be evaluated when enough data is available for drawing conclusions.

Further plans for VSUKE consists of making the program more accessible for use in quality control. First and foremost this consists of choosing the right warning limits. It is expected, however, that by running the SPC experiment such limits will become evident. Later on VSUKE statistics should be sent automatically to the users, perhaps by directly running the output from the program into the fax.

However, in order to put the results in a wider context and prepare for the next set of experiments, literature in the field of research should be investigated in case there are methods and concepts that should be included before new experiments are designed.

References

- [1] DNMI-Report no. 06/94 KLIBAS "VSUKE, NSUKE og VSDUMP: Datautskrift fra arbeidslager.", Oslo, March 4th, 1994.
- [2] DNMI-Report no. 04/95 KLIBAS "Revisjon av data-ut-programmene RRUTM og VSUKE.", Oslo, January 10th, 1995.
- [3] DNMI-Report no. 48/97 KLIBAS "Ukesoversikter fra TELE med programmet VSUKE v.2.0.", Oslo, March 22nd, 1997.

- [4] *DNMI-Report no. 26/99 KLIBAS "Using VSUKE v.2.1 in Quality Control."*, Oslo, April 5th, 1999.
- [5] *DNMI-Report no. 12/99 KLIMA "Quality Control of Climate Data: An Automatic Approach"*, Oslo, March 19th, 1999.

Forcing old tasks on the WBS list in the DRIFT program to be managed by the KLIBAS database system

Petter Øgland

Norwegian Meteorological Institute
Niels Henrik Abels vei 40
P.O. Box 43, Blindern
N-0313 Oslo, NORWAY
(subm. April 6, 1999)

ABSTRACT

The purpose of the computer program DRIFT is provide statistics for helping manage computer program development of the KLIBAS database system as a Software Process Improvement task. The program collects log from the KLIBAS system and generates statistics in order to assist the improvement circle. This note addresses a problem with the program causing it to top the problem statistics in the KLIBAS maintenance system.

The DRIFT program

September 1998: The DRIFT program [1] 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.

April 1999: In the version 1.1 of DRIFT, [2], both the DRIFT program and KLIBAS/NORDKLIM progress is monitored by a revised type of statistical process control (SPC), based on counting the number of tasks not yet being solved according to Work Breakdown Structure (WBS). Statistics will be used in the monthly status reports.

Each run by DRIFT is logged, registering program breakdown as a defect. The first log of the program is in December 1997. It was then run 12 times, failing 2 of these (16.7%). 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.

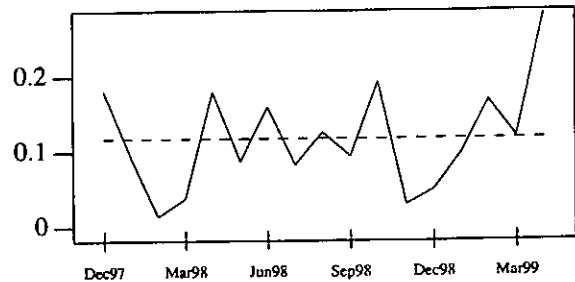


Figure 1. Relative number of defects

The average value dashed in the figure is 0.10. Statistics of this kind is present method for documenting general system quality and improvement.

In order to focus not only on long term problems caused by breakdown of the program, a circular structure of the last twelve months is used for recording errors and warnings from the program.

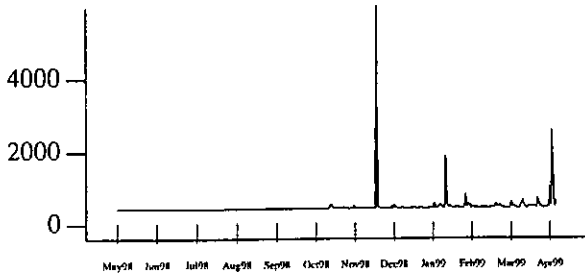


Figure 2. Problems day by day

The type of curve represented by figure 2 is used for checking whether there have been much update of the program recently or if the program is systematically reporting a great number of problems.

The current problem

The message "Error: list of programs in DRIFT, chapter 12, is not a list of programs on the agenda" was first recorded on an err file on Fri Mar 19 1999. We ask ourselves what can be learned about the KLIBAS system and DRIFT by understanding this problem, and how may it help achieving the ultimate goal of the system?

The ultimate goal of the KLIBAS system is to be totally self-contained in terms of automatically carrying out all operations of quality control of climate data with as little manual interaction as possible [3].

In order both to understand the problem and to solve it, a plan must be devised, an experiment must be carried out and the results must be interpreted.

Planning an experiment

The purpose of the computer program DRIFT is provide statistics for helping manage computer program development of the KLIBAS database system as a Software Process Improvement task. The program collects log from the KLIBAS system and generates statistics in order to assist the improvement circle.

The figure shows the day by day count of unsolved tasks in the NORDKLIM/KLIBAS project. The WBS (Work Breakdown Structure) for KLIBAS was redefined on Sat Apr 3 1999 by coordinating it with a preliminary NORDKLIM WBS. The plot is handled as with Statistical Process Control (SPC) by also plotting dashed control lines and process capability in terms of six times the standard deviation of the process.

The most recent problem with the DRIFT system is a failure in a presentation of tasks on the Work Breakdown Structure (WBS) for NORDKLIM. The list of ten most critical tasks on the WBS does not correspond with the actual WBS. The problem should be solved in a manner which would make future problems visible of the SPC curve below. The DRIFT program should force the oldest problems onto the present agenda by pushing the problem to the DRIFT error list.

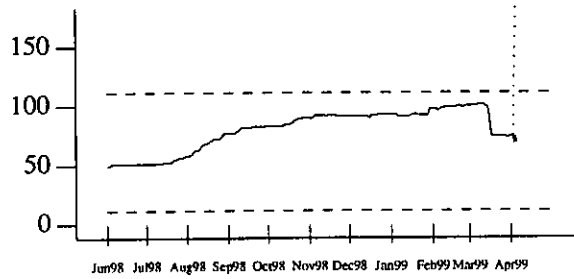


Figure 3. DRIFT evaluation by SPC

Running the experiment

According to specifications above, an experiment should be run and evaluated. The actual running of the experiment should be handled by the systematic running of the DRIFT program itself.

The DRIFT program is run on a daily basis by KAPO in order to assist in deciding which job of the day that should be carried out. It is also run automatically twice a day by crontab via KAPO.

When the system has been running for sufficiently long time, it should again be evaluated, this note being a vital reference for documenting improvements.

Conclusion

As a step in solving the title problem in this note, an experiment has been designed, and measures on how to carry this experiment out have been described. The results will be evaluated when enough data is available for drawing conclusions.

The plan is to make overdue tasks on the WBS into problems on the DRIFT err files which will either cause redefinition of the WBS, the DRIFT program or force the problems to be solved.

However, in order to put the results in a wider context and prepare for the next set of experiments, literature in the field of research

should be investigated in case there are methods and concepts that should be included before new experiments are designed.

References

- [1] *DNMI-Report no. 47/98 KLIBAS "Towards an ISO-9000 standard for system development by DRIFT."*, Oslo, September 7th, 1998.
- [2] *DNMI-Report no. 25/99 KLIBAS "Automatic update of NORDKLIM project plans by DRIFT v.1.1."*, Oslo, April 3rd, 1999.
- [3] *DNMI-Report no. 12/99 KLIMA "Quality Control of Climate Data: An Automatic Approach"*, Oslo, March 19th, 1999.

Automatically updating the KLIBAS web page

Petter Øgland

Norwegian Meteorological Institute
Niels Henrik Abels vei 40
P.O. Box 43, Blindern
N-0313 Oslo, NORWAY
(subm. April 6, 1999)

ABSTRACT

The purpose of the program WEB is to automatically update the html pages describing the current list of KLIBAS reports (<http://gust/>). The WEB program is designed be run both on its own and as a part of the KAPO system. This note addresses a problem with the program causing it to top the problem statistics in the KLIBAS maintenance system.

The WEB program

April 1999: The first version of the WEB program [1] aims at producing a simplified overview of KLIBAS reports, using crontab scheduling both on GUST and GALE in order to produce and transfer the latest html list of reports.

Each run by WEB is logged, registering program breakdown as a defect. The first log of the program is in December 1998. It was then run 31 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.

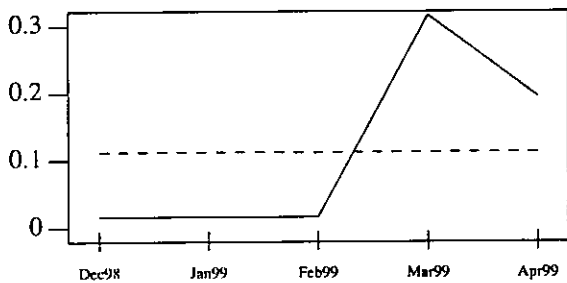


Figure 1. Relative number of defects

The average value dashed in the figure is 0.10. Statistics of this kind is present method for documenting general system quality and improvement.

In order to focus not only on long term problems caused by breakdown of the program, a circular structure of the last twelve months is used for recording errors and warnings from the program.

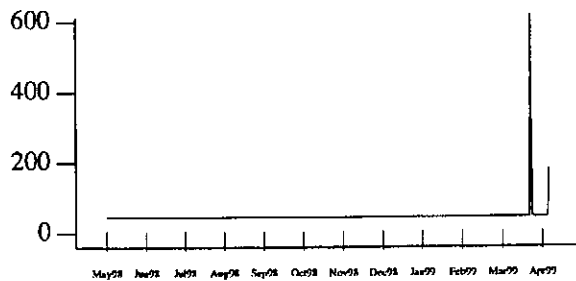


Figure 2. Problems day by day

The type of curve represented by figure 2 is used for checking whether there have been much update of the program recently or if the program is systematically reporting a great number of problems.

The current problem

The message "Error: rcp should be done from GALEHA, not GUS" was first recorded on an err file on Mon Mar 29 1999. We ask ourselves what can be learned about the KLIBAS system and WEB by understanding this problem, and how may it help achieving the ultimate goal of the system?

The ultimate goal of the KLIBAS system is to be totally self-contained in terms of automatically carrying out all operations of quality control of climate data with as little manual interaction as possible [2].

In order both to understand the problem and to solve it, a plan must be devised, an experiment must be carried out and the results must be interpreted.

Planning an experiment

The purpose of the program WEB is to automatically update the html pages describing the current list of KLIBAS reports (<http://gust/>). The WEB program is designed to run both on its own and as a part of the KAPO system.

The figure shows the day by day count of reports on the web page for the current year minus the number of days passed in the current year. This statistic is meant to work as a SPC variable for monitoring the WEB program. If the variable is out of limits too few reports are being written. The plot is a part of the Statistical Process Control (SPC) for the system and consists of a dashed line representing the average value of the day for the process and the upper limit of the process (process capability) given in terms of six times the standard deviation of the process.

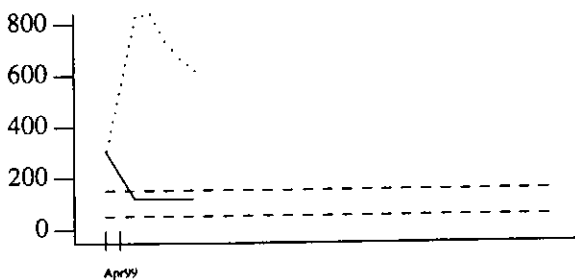


Figure 3. WEB evaluation by SPC

Running the experiment

According to specifications above, an experiment should be run and evaluated. The actual running of the experiment should be handled by the systematic running of the WEB program itself.

The WEB program is run automatically twice a day as scheduled by crontab and normally several times a day as well when the KAPO program is run manually. The WEB program produces a html file called Rapport98.html containing the complete list of KLIBAS reports from 1993 up to the present.

When the system has been running for sufficiently long time, it should again be evaluated, this note being a vital reference for documenting improvements.

Conclusion

As a step in solving the title problem in this note, an experiment has been designed, and measures on how to carry this experiment out have been described. The results will be evaluated when enough data is available for drawing conclusions.

The plan is to monitor the web page. Other statistics measuring the quality of the page may be considered later on when the capability for the process in terms of the present statistic has been found and the process is under control.

However, in order to put the results in a wider context and prepare for the next set of experiments, literature in the field of research should be investigated in case there are methods and concepts that should be included before new experiments are designed.

References

- [1] DNMI-Report no. 27/99 KLIBAS "Automatic update of the KLIBAS web-page by WEB v.1.0.", Oslo, April 6th, 1999.
- [2] DNMI-Report no. 12/99 KLIMA "Quality Control of Climate Data: An Automatic Approach", Oslo, March 19th, 1999.

Improving on a temporary solution for the VIND_REG program in the the XVIND routine at DNMI

Petter Øgland

Norwegian Meteorological Institute
Niels Henrik Abels vei 40
P.O. Box 43, Blindern
N-0313 Oslo, NORWAY
(subm. April 7, 1999)

ABSTRACT

The purpose of the program VIND_REG is to insert AWS wind registrations from diskette into the KLIBAS datatable VIND and the A-tables. Each time the program is executed, performance statistics are added to the log files and the performance statistics file of the month is updated. This note addresses a problem with the program causing it to top the problem statistics in the KLIBAS maintenance system.

The VIND_REG program

July 1998: The first version of the system [1] is programmed according to specifications as written September 1997.

August 1998: Specifications for the VIND-DEK program [2] were made by Lars Andresen and Per Ove Kjensli. The program is intended to constitute the main part of the VIND_REG weather data routine at the Climatology Division.

August 1998: In the revised version 1.1 of VIND_REG [3] minor faults in the algorithm have been corrected. The documentation has been updated and partly rewritten based on handwritten comments on report no. 22/98 KLIBAS.

October 1998: What is new in the 1.2 version of VIND_REG [4] as compared with the old version 1.1 is that the program is now running systematically every hour of the day by the crontab. The program is integrated with a quality check routine using the program VINDDEK, and the part of the program that is generating statistics for daily monitoring and monthly printing in the KLIBAS statistics report has been improved. In addition to storing observations in separate A-tables, the program now also stores data in a general VIND-table preparing the routine for the new Oracle8 database structure. The report contains a definition of the XVIND data table in the

appendix.

April 1999: A number of minor adjustments have been done on the road from version 1.2 to the version 1.3 of VIND_REG. [5]. The latest feature is added control statistics measuring the total number of observations recorded for the present month. These kind of measurements are done on a daily basis and should help monitoring that the routine is functioning as required.

Each run by VIND_REG is logged, registering program breakdown as a defect. 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.

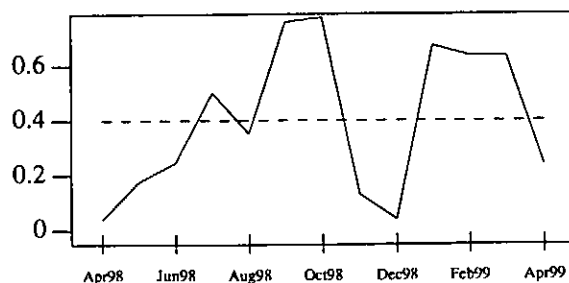


Figure 1. Relative number of defects

The average value dashed in the figure is 0.3616. Statistics of this kind is present method for documenting general system quality and improvement.

In order to focus not only on long term problems caused by breakdown of the program, a circular structure of the last twelve months is used for recording errors and warnings from the program.

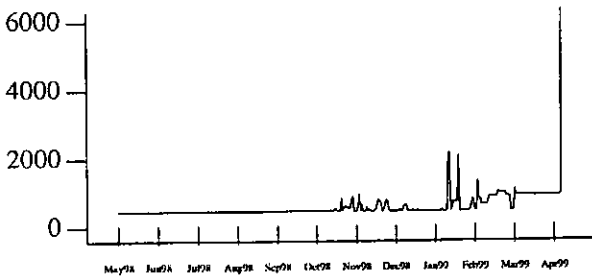


Figure 2. Problems day by day

The type of curve represented by figure 2 is used for checking whether there have been much update of the program recently or if the program is systematically reporting a great number of problems.

The current problem

The message *"The error for VIND_REG on Wed Feb 24 1999 should be solved in a better manner"* was first recorded on an err file on Mon Mar 29 1999. We ask ourselves what can be learned about the KLIBAS system and VIND_REG by understanding this problem, and how may it help achieving the ultimate goal of the system?

The ultimate goal of the KLIBAS system is to be totally self-contained in terms of automatically carrying out all operations of quality control of climate data with as little manual interaction as possible [6].

In order both to understand the problem and to solve it, a plan must be devised, an experiment must be carried out and the results must be interpreted.

Planning an experiment

The purpose of the program VIND_REG is to insert AWS wind registrations from diskette into the KLIBAS datatable VIND and the A-tables. Each time the program is executed, performance statistics are added to the log files and the performance statistics file of the month is

updated.

The figure shows the day by day count of rows for the relevant month in the XVIND data table. This statistic is meant to work as a SPC variable for monitoring the VIND_REG program. If the variable is out of limits, probably the routine is not working as required. The plot is a part of the Statistical Process Control (SPC) for the system and consists of a dashed line representing the average value of the day for the process and the upper limit of the process (process capability) given in terms of six times the standard deviation of the process.

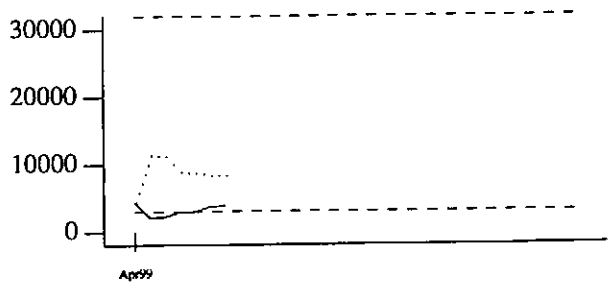


Figure 3. VIND_REG evaluation by SPC

Running the experiment

According to specifications above, an experiment should be run and evaluated. The actual running of the experiment should be handled by the systematic running of the VIND_REG program itself.

As diskettes arrive in the beginning of every month they are copied to the tmp-directory `~kabase/vindreg`. Every one hour this directory is checked by the VIND_REG program via crontab and any new file on the directory is loaded into the KLIBAS database tables. Status from the VIND_REG program is stored on the `vind_reg.txt` file.

When the system has been running for sufficiently long time, it should again be evaluated, this note being a vital reference for documenting improvements.

Conclusion

As a step in solving the title problem in this note, an experiment has been designed, and measures on how to carry this experiment out have been described. The results will be evaluated when enough data is available for drawing conclusions.

The plan is to monitor the `vind_reg` routine by use of SPC techniques, using the present parameter as a first approach. As we gain experience with this parameter, more sophisticated methods may be explored. The preliminary solution for checking observations done on the 24th of February 1999, and documented in the latest version of the system documentation, will be handled when the SPC stabilises.

However, in order to put the results in a wider context and prepare for the next set of experiments, literature in the field of research should be investigated in case there are methods and concepts that should be included before new experiments are designed.

References

- [1] *DNMI-Report no. 22/98 KLIBAS "Computer program VIND_REG."*, Oslo, July 12th, 1998.
- [2] *DNMI-Report no. 39/98 KLIBAS "Computer program VINDDEK."*, Oslo, August 18th, 1998.
- [3] *DNMI-Report no. 43/98 KLIBAS "Computer program VIND_REG v.1.1."*, Oslo, August 27th, 1998.
- [4] *DNMI-Report no. 56/98 KLIBAS "Storing AWS wind observations by use of VIND_REG v.1.2."*, Oslo, October 20th, 1998.
- [5] *DNMI-Report no. 28/99 KLIBAS "Adding process control features in VIND_REG v.1.3."*, Oslo, April 7th, 1999.
- [6] *DNMI-Report no. 12/99 KLIMA "Quality Control of Climate Data: An Automatic Approach"*, Oslo, March 19th, 1999.

Daily quality evaluation for automatic weather stations by AUTO_MKK v.1.1

Petter Øgland

Norwegian Meteorological Institute
Niels Henrik Abels vei 40
P.O. Box 43, Blindern
N-0313 Oslo, NORWAY
(subm. April 8, 1999)

ABSTRACT

The purpose of the program AUTO_MKK is to make a summary of quality control statistics from data in ALA. The program is responsible for sending the summary to ITAS, IA and Klima by e-mail. This note addresses a problem with the program causing it to top the problem statistics in the KLIBAS maintenance system.

The AUTO_MKK program

May 1995: The first version of the program MKK [1] was established in order to produce monthly quality evaluation reports for the automatic weather stations at DNMI.

July 1997: The AUTO_INN system [2] was investigated and partly reprogrammed from scratch. One of the new features in version 1.1 was a daily quality evaluation called AUTO_MKK, producing the same type of results as the monthly orientated MKK.

September 1998: The version 2.0 of MKK is significantly restructured from the initial version of the program [3] described in KLIBAS report no. 15/95. The present system is fully explained in this report. Among the latest additions to the MKK system are tables displaying monthly values of UU, TT, P0 and RR_12 sorted by standard deviation.

April 1999: In the version 1.1 of AUTO_MKK [4] the program is developed further, now automatically updating a list of the ten worst stations according to ADK quality control. An error in the list of stations not operative was corrected. It is expected that the SPC count of problems should increase slightly with this version. In this version standard SPC parameters have also been included in the defects plot.

Each run by AUTO_MKK is logged, registering program breakdown as a defect. The first log of the program is in July 1997. It was then run 104 times, failing 3 of these (2.9%). 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.

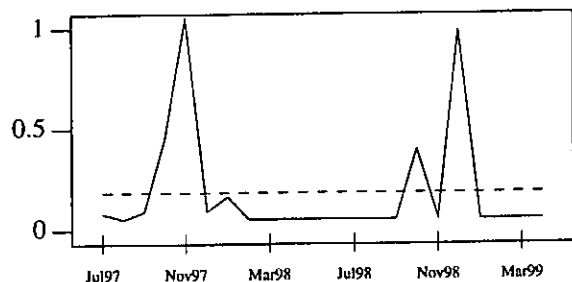


Figure 1. Relative number of defects

The average value dashed in the figure is 0.13. Statistics of this kind is present method for documenting general system quality and improvement.

In order to focus not only on long term problems caused by breakdown of the program, a circular structure of the last twelve months is used for recording errors and warnings from the program.

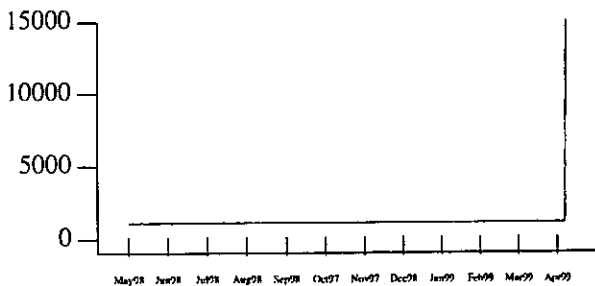


Figure 2. Problems day by day

The type of curve represented by figure 2 is used for checking whether there have been much update of the program recently or if the program is systematically reporting a great number of problems.

The current problem

The message "Error: The list of missing stations on AWS mail (AUTO_MKK) is not working" was first recorded on an err file on Sat Feb 20 1999. We ask ourselves what can be learned about the KLIBAS system and AUTO_MKK by understanding this problem, and how may it help achieving the ultimate goal of the system?

The ultimate goal of the KLIBAS system is to be totally self-contained in terms of automatically carrying out all operations of quality control of climate data with as little manual interaction as possible [5].

In order both to understand the problem and to solve it, a plan must be devised, an experiment must be carried out and the results must be interpreted.

Planning an experiment

The purpose of the program AUTO_MKK is to make a summary of quality control statistics from data in ALA. The program is responsible for sending the summary to ITAS, IA and Klima by e-mail.

The plot shows the day by day problems recorded by reading quality control logs from the ADK system on a daily basis and recording the number of problems for each station for the last three days.

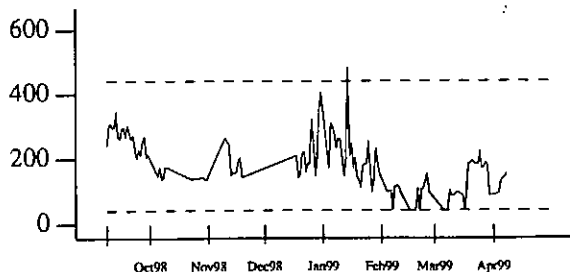


Figure 3. AUTO_MKK evaluation by SPC

Running the experiment

According to specifications above, an experiment should be run and evaluated. The actual running of the experiment should be handled by the systematic running of the AUTO_MKK program itself.

The AUTO_MKK program is run automatically once a day by crontab via the AUTO_INN program. It makes use of output from the MND2ALA and ALA2TELE programs, but the output from the ADK program is of greatest importance. A daily quality evaluation is written to file (auto_mkk.mail) and distributed to IA, KLIMA and ITAS by e-mail.

When the system has been running for sufficiently long time, it should again be evaluated, this note being a vital reference for documenting improvements.

Conclusion

As a step in solving the title problem in this note, an experiment has been designed, and measures on how to carry this experiment out have been described. The results will be evaluated when enough data is available for drawing conclusions.

The plan for further improvement of the AUTO_MKK program consists of including standard SPC parameters such as averages, process capability and warning limits in the plot in order to find out how many errors we expect the system to have and when special attention have to be made as process limits are exceeded.

However, in order to put the results in a wider context and prepare for the next set of experiments, literature in the field of research should be investigated in case there are methods and concepts that should be included before new experiments are designed.

References

- [1] *DNMI-Report no. 15/95 KLIBAS "MKK: Månedlig kvalitetskontroll av automatstasjonsdata."*, Oslo, May 11th, 1995.
- [2] *DNMI-Report no. 59/97 KLIBAS "Innlesing AUTO_INN v.1.1 for AVS: Programmer mnd2ata, mnd2hla, ala2tele, adk og mkk."*, Oslo, July 29th, 1997.
- [3] *DNMI-Report no. 49/98 KLIBAS "Monthly AWS Quality Control Statistics MKK v.2.0."*, Oslo, September 9th, 1998.
- [4] *DNMI-Report no. 29/99 KLIBAS "Daily Quality Evaluation of AWS by the program AUTO_MKK v.1.1."*, Oslo, April 8th, 1999.
- [5] *DNMI-Report no. 12/99 KLIMA "Quality Control of Climate Data: An Automatic Approach"*, Oslo, March 19th, 1999.

Producing statistics for daily monitoring of the AANDERAA routine

Petter Øgland

Norwegian Meteorological Institute
Niels Henrik Abels vei 40
P.O. Box 43, Blindern
N-0313 Oslo, NORWAY
(subm. April 9, 1999)

ABSTRACT

The purpose of the program AANDERAA_INN is to run and monitor the transport of weather data from datafiles in the RUTINE3 PC system to the data tables in the KLIBAS Aanderaa AWS system. This note addresses a problem with the program causing it to top the problem statistics in the KLIBAS maintenance system.

The AANDERAA_INN program

October 1986: The original system [1] for quality control and data treatment for AANDERAA AWS was established.

June 1987: A manual [2] was written describing how to use observations from Aanderaa automatic weather stations at DNMI.

April 1994: The report [3] contains a plan on how to migrate the data processing routine for non-realtime automatic weather stations (Aanderaa) from the ND system to the KLIBAS climatological database system on the SGI computer.

October 1994: This is a description of the first version of the data collection and quality control module for the non-realtime automatic weather stations (Aanderaa weather stations). The report [4] contains technical descriptions and guide on how to run the system.

February 1995: The first version of the program [5] that inserts AANDERAA observations into the KLIBAS datatables was constructed.

February 1995: The report [6] describes a prototype for loading historical data into Oracle datatables in the case of Aanderaa AWS was established.

February 1995: This report [7] was constructed as a folder designed to contain all software documentation being developed for the AANDERAA AWS system, including modules for data collection, quality control and programs

for producing weather statistics.

April 1996: This reports [8] gives a status on all currently working routines that constitute the Aanderaa AWS system at the time of writing. Most of the reports describes the data collection programs.

October 1996: A first step in order to move Aanderaa observations from the database on the ND computer over to the KLIBAS Oracle database system [9] is described.

December 1997: The report [10] describes a program ALA2HLA was constructed in order to copy data from the aanderaa working storage AANDERAA to each of the particular data tables in the final storage HLA.

April 1999: As a first approach to automatic monitoring of the AANDERAA data routine certain curves and statistics have been added to the AANDERAA_INN [11] computer program, counting the number of rows in the AANDERAA datatable and listing the stations in the table whenever the program is being run. The statistics will provide information for automatic maintenance of the AANDERAA datatable.

Each run by AANDERAA_INN is logged, registering program breakdown as a defect. The first log of the program is in January 1996. It was then run 41 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.

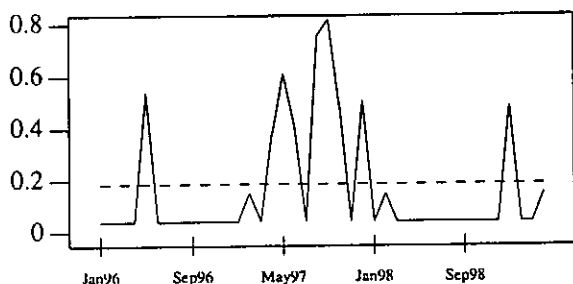


Figure 1. Relative number of defects

The average value dashed in the figure is 0.15. Statistics of this kind is present method for documenting general system quality and improvement.

The current problem

The message "Error: some kind of monitoring of the AANDERAA routine need to be done in order to make sure that it is" was first recorded on an err file on Sat Feb 20 1999. We ask ourselves what can be learned about the KLIBAS system and AANDERAA_INN by understanding this problem, and how may it help achieving the ultimate goal of the system?

The ultimate goal of the KLIBAS system is to be totally self-contained in terms of automatically carrying out all operations of quality control of climate data with as little manual interaction as possible [12].

In order both to understand the problem and to solve it, a plan must be devised, an experiment must be carried out and the results must be interpreted.

Planning an experiment

The purpose of the program AANDERAA_INN is to run and monitor the transport of weather data from datafiles in the RUTINE3 PC system to the data tables in the KLIBAS Aanderaa AWS system.

The figure shows the day by day count of rows for the relevant month in the AANDERAA data table. This statistic is meant to work as a SPC variable for monitoring the AANDERAA_INN program. If the variable is out of limits, the routine is probably not working as required. The plot is a part of the Statistical Process Control (SPC) for the system and consists of a dashed line representing the average value of the

day for the process and the statistical limits of the process (process capability) given in terms of six times the standard deviation of the process.

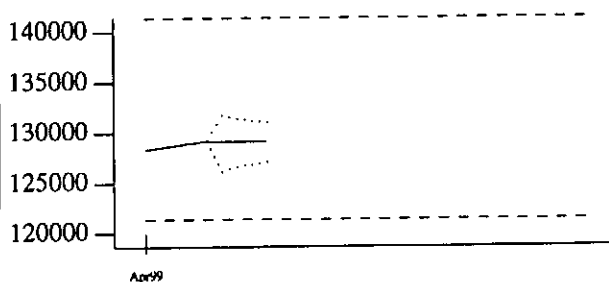


Figure 2. AANDERAA_INN evaluation by SPC

Running the experiment

According to specifications above, an experiment should be run and evaluated. The actual running of the experiment should be handled by the systematic running of the AANDERAA_INN program itself.

The AANDERAA_INN program is used to monitor the data collection routine defined by RUTINE3 on the PC-Aanderaa routine and the datainn.csh program in KLIBAS. AANDERAA_INN is run by RUTINE3 when needed, not by crontab. AANDERAA_INN then starts the datainn.sh program which reads data from data file and inserts the observations into the AANDERAA data table.

When the system has been running for sufficiently long time, it should again be evaluated, this note being a vital reference for documenting improvements.

Conclusion

As a step in solving the title problem in this note, an experiment has been designed, and measures on how to carry this experiment out have been described. The results will be evaluated when enough data is available for drawing conclusions.

The plan is to monitor the AANDERAA routine by use of SPC techniques, using the present parameter as a first approach. As we gain experience with this parameter, more sophisticated methods may be explored. Later on automatic updates may be done on the AANDERAA data table, deleting or restructuring the table when statistics indicate that the system is getting out of control.

However, in order to put the results in a wider context and prepare for the next set of experiments, literature in the field of research should be investigated in case there are methods and concepts that should be included before new experiments are designed.

References

- [1] *DNMI-Report no. 52/86 KLIMA "Automat-stasjoner. Databearbeiding."*, Oslo, October 6th, 1986.
- [2] *DNMI-Report no. 21/87 KLIMA "Praktisk bruk av data fra automat-stasjoner (Aanderaa stasjoner)."*, Oslo, June 2nd, 1987.
- [3] *DNMI-Report no. 09/94 KLIBAS "Omlegging av EDB-rutiner for Aanderaa-stasjoner: Spesifikasjonsrapport."*, Oslo, April 18th, 1994.
- [4] *DNMI-Report no. 22/94 KLIBAS "Data-kontroll for Aanderaa-stasjoner. PD-rutinen."*, Oslo, October 5th, 1994.
- [5] *DNMI-Report no. 09/95 KLIBAS "Innlasting av Aanderaa-data til arbeidslager på Typhoon."*, Oslo, February 6th, 1995.
- [6] *DNMI-Report no. 10/95 KLIBAS "Rutine for innlasting av historiske Aanderaa-data til hovedlager på Typhoon."*, Oslo, February 20th, 1995.
- [7] *DNMI-Report no. 11/95 KLIBAS "Programvare for Aanderaa-rutinen ved Klimaavdelingen."*, Oslo, February 20th, 1995.
- [8] *DNMI-Report no. 11/96 KLIBAS "Beskrivelse av rutine for ikke-telegraferende automater (Aanderaa-stasjoner)."*, Oslo, April 13th, 1996.
- [9] *DNMI-Report no. 14/96 KLIBAS "Overflytting og delvis verifikasjon av historiske Aanderaa data til Oracle."*, Oslo, October 19th, 1996.
- [10] *DNMI-Report no. 63/97 KLIBAS "AL2HLA - Automatic Update of Datatables HLA from Datatable AANDERAA."*, Oslo, December 18th, 1997.
- [11] *DNMI-Report no. 30/99 KLIBAS "Using AANDERAA_INN v.1.1 for monitoring AWS data collection."*, Oslo, April 9th, 1999.
- [12] *DNMI-Report no. 12/99 KLIMA "Quality Control of Climate Data: An Automatic Approach"*, Oslo, March 19th, 1999.

Adding a control feature in CHECK_MAIL in order to make sure that the AUTO_MKK program is running

Petter Øgland

Norwegian Meteorological Institute
Niels Henrik Abels vei 40
P.O. Box 43, Blindern
N-0313 Oslo, NORWAY
(subm. April 13, 1999)

ABSTRACT

The program CHECK_MAIL was constructed in order to automatically delete old mails for system user kabase. If not deleted automatically, the mailbox of kabase has a tendency of growing far to full before mails are deleted manually. The problem has been noted on the monthly list of problems (as "logging av backup") since September 1995. This note addresses a problem with the program causing it to top the problem statistics in the KLIBAS maintenance system.

The CHECK_MAIL program

February 1998: The program CHECK_MAIL [1] was designed for automatically deleting redundant mail for system user KABASE.

February 1999: The first documented version of the MAILSTAT program [2] so far only collects statistics from mailbox. Further programming will be needed for presenting the statistics as more values are collected. The program in its present form is important, however, as an aid for understanding how the KLIBAS warning mail system performs as priority lists for maintenance and development is based on this system.

April 1999: As it was discovered that the original CHECK_MAIL system [3] had not been running for several months, allowing the mailbox for KABASE to grow uncontrollably large, it was apparent that the program needed better monitoring. In the version 1.1 of CHECK_MAIL, it updates a daily SPC chart and an automatic mail communication between the users KABASE and KAPO is established.

Each run by CHECK_MAIL is logged, registering program breakdown as a defect. The first log of the program is in February 1998. It was then run 52 times, failing 10 of these (19.2%).

The number of users during this first month was 2. The curve below shows the relative number of runs of the program that has been failing according to the log.

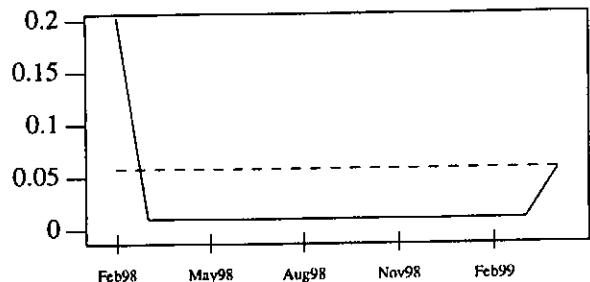


Figure 1. Relative number of defects

The average value dashed in the figure is 0.05. Statistics of this kind is present method for documenting general system quality and improvement.

The current problem

The message "Error: make the CHECK_MAIL program remove KLIBAS warnings from the kapo mailbox" was first recorded on an err file on Wed Mar 31 1999. We ask ourselves what can be learned about the KLIBAS system and CHECK_MAIL by understanding this

problem, and how may it help achieving the ultimate goal of the system?

The ultimate goal of the KLIBAS system is to be totally self-contained in terms of automatically carrying out all operations of quality control of climate data with as little manual interaction as possible [4].

In order both to understand the problem and to solve it, a plan must be devised, an experiment must be carried out and the results must be interpreted.

Planning an experiment

The program CHECK_MAIL was constructed in order to automatically delete old mails for system user kabase. If not deleted automatically, the mailbox of kabase has a tendency of growing far to full before mails are deleted manually. The problem has been noted on the monthly list of problems (as "logging av backup") since September 1995.

The figure shows the day by day count of letters in the kabase mailbox on GALE. The plot is a part of the Statistical Process Control (SPC) for the system and consists of a dashed line representing the average value of the day for the process and the upper limit of the process (process capability) given in terms of six times the standard deviation of the process.

As CHECK_MAIL has a special responsibility to check that system mail is arriving as expected, the SPC need to be further developed. The presently chosen limits in the curve below are chosen in order to included the expected amount of AWS STATUS messages from the AUTO_MKK program. If not enough of messages are received, the SPC goes out of control.

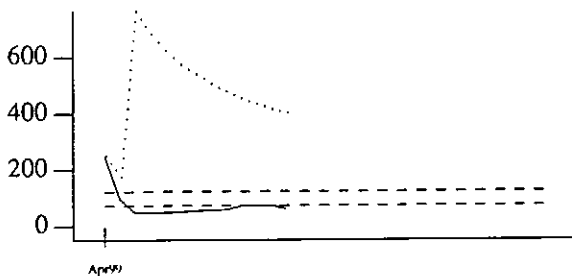


Figure 2. CHECK_MAIL evaluation by SPC

Running the experiment

According to specifications above, an experiment should be run and evaluated. The actual running of the experiment should be handled by the systematic running of the CHECK_MAIL program itself.

The CHECK_MAIL program is run automatically once a day by the user KABASE on GALE as schedules by crontab. The program is responsible for deleting excess mail in the kabase mailbox and to communicate to the user KAPO that everything is okay. The communication is done via email so if KAPO and MAILSTAT were to be located on a random computer, not connected to GALE in anyway, the users still communicated and may be able to give each other information and instructions.

When the system has been running for sufficiently long time, it should again be evaluated, this note being a vital reference for documenting improvements.

Conclusion

As a step in solving the title problem in this note, an experiment has been designed, and measures on how to carry this experiment out have been described. The results will be evaluated when enough data is available for drawing conclusions.

As a plan for further development, it is needed to give a more detailed analysis of the letters that arrive in the kabase mailbox. The letters should be grouped, counted and necessary adjustments should be executed when the systems seems to get out of control.

However, in order to put the results in a wider context and prepare for the next set of experiments, literature in the field of research should be investigated in case there are methods and concepts that should be included before new experiments are designed.

References

- [1] DNMI-Report no. 06/98 KLIBAS "Computer program CHECK_MAIL.", Oslo, February 11th, 1998.
- [2] DNMI-Report no. 13/99 KLIBAS "Monitoring KLIBAS system mail by MAILSTAT v.1.0.", Oslo, February 22nd, 1999.
- [3] DNMI-Report no. 24/99 KLIBAS "Adding SPC and communication features in CHECK_MAIL v.1.1.", Oslo, April 2nd,

Adding features in MAILSTAT in order to remove warning messages and check up on the CHECK_MAIL program

Petter Øgland

Norwegian Meteorological Institute
Niels Henrik Abels vei 40
P.O. Box 43, Blindern
N-0313 Oslo, NORWAY
(subm. April 14, 1999)

ABSTRACT

The purpose of the computer program MAILSTAT is to count the number of mails arriving daily and do some simple pattern recognition in the mailbox in order to get a clearer picture of how the KLIBAS mail warning system is performing. This note addresses a problem with the program causing it to top the problem statistics in the KLIBAS maintenance system.

The MAILSTAT program

February 1998: The program CHECK_MAIL [1] was designed for automatically deleting redundant mail for system user KABASE.

September 1998: The DRIFT program [2] 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.

February 1999: The first documented version of the MAILSTAT program [3] so far only collects statistics from mailbox. Further programming will be needed for presenting the statistics as more values are collected. The program in its present form is important, however, as an aid for understanding how the KLIBAS warning mail system performs as priority lists for maintenance and development is based on this system.

April 1999: As it was discovered that the original CHECK_MAIL system [4] had not been running for several months, allowing the mailbox for KABASE to grow uncontrollably large, it was apparent that the program needed better monitoring. In the version 1.1 of CHECK_MAIL, it updates a daily SPC chart and an automatic mail communication between the users KABASE and

KAPO is established.

April 1999: In the version 1.1 of DRIFT, both the DRIFT program and KLIBAS/NORD-KLIM [5] progress is monitored by a revised type of statistical process control (SPC), based on counting the number of tasks not yet being solved according to Work Breakdown Structure (WBS). Statistics will be used in the monthly status reports.

April 1999: The version 2.0 of MAILSTAT [6] was moved from the KLIBAS database server GALEHA to the local SGI computer HALO as it operates on the local mailing system used by kapo. System mail on GALEHA is handled by the system user kabase by using the CHECK_MAIL system. The mail statistics in version 2.0 of MAILSTAT have been redesigned in order to communicate with the CHECK_MAIL system.

Each run by MAILSTAT is logged, registering program breakdown as a defect. The first log of the program is in November 1997. It was then run 13 times, failing 1 of these (7.7%). 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.

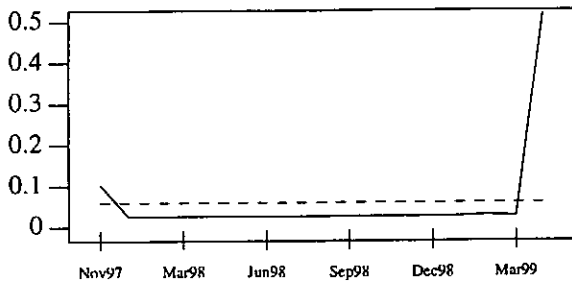


Figure 1. Relative number of defects

The average value dashed in the figure is 0.03. Statistics of this kind is present method for documenting general system quality and improvement.

The current problem

The message "Error: make the MAILSTAT program remove KLIBAS warnings from the kapo mailbox" was first recorded on an err file on Wed Mar 31 1999. We ask ourselves what can be learned about the KLIBAS system and MAILSTAT by understanding this problem, and how may it help achieving the ultimate goal of the system?

The ultimate goal of the KLIBAS system is to be totally self-contained in terms of automatically carrying out all operations of quality control of climate data with as little manual interaction as possible [7].

In order both to understand the problem and to solve it, a plan must be devised, an experiment must be carried out and the results must be interpreted.

Planning an experiment

The purpose of the computer program MAILSTAT is to count the number of mails arriving daily and do some simple pattern recognition in the mailbox in order to get a clearer picture of how the KLIBAS mail warning system is performing.

The figure shows the day by day count of letters in the kapo mailbox on HALO. The plot is a part of the Statistical Process Control (SPC) for the system and consists of a dashed line representing the average value of the day for the process and the upper limit of the process (process capability) given in terms of six times the standard deviation of the process.

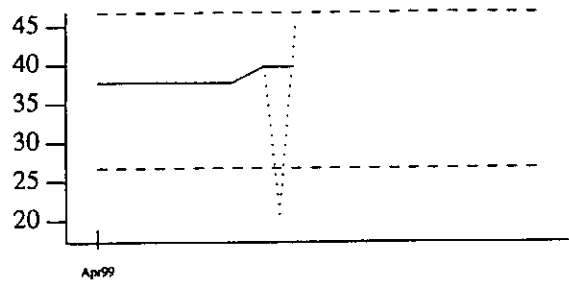


Figure 2. MAILSTAT evaluation by SPC

Running the experiment

According to specifications above, an experiment should be run and evaluated. The actual running of the experiment should be handled by the systematic running of the MAILSTAT program itself.

The MAILSTAT program is run automatically every hour of the day as schedules by crontab. The program is responsible for checking the mailbox once a day in order to get a profile on how the KLIBAS warning mail system is performing. Warnings are deleted.

When the system has been running for sufficiently long time, it should again be evaluated, this note being a vital reference for documenting improvements.

Conclusion

As a step in solving the title problem in this note, an experiment has been designed, and measures on how to carry this experiment out have been described. The results will be evaluated when enough data is available for drawing conclusions.

As a plan for further development, it is needed to give a more detailed analysis of the letters that arrive in the kapo mailbox. The letters should be grouped, counted and necessary adjustments should be executed when the systems seems to get out of control.

However, in order to put the results in a wider context and prepare for the next set of experiments, literature in the field of research should be investigated in case there are methods and concepts that should be included before new experiments are designed.

References

- [1] *DNMI-Report no. 06/98 KLIBAS "Computer program CHECK_MAIL."*, Oslo, February 11th, 1998.
- [2] *DNMI-Report no. 47/98 KLIBAS "Towards an ISO-9000 standard for system development by DRIFT."*, Oslo, September 7th, 1998.
- [3] *DNMI-Report no. 13/99 KLIBAS "Monitoring KLIBAS system mail by MAILSTAT v.1.0."*, Oslo, February 22nd, 1999.
- [4] *DNMI-Report no. 24/99 KLIBAS "Adding SPC and communication features in CHECK_MAIL v.1.1."*, Oslo, April 2nd, 1999.
- [5] *DNMI-Report no. 25/99 KLIBAS "Automatic update of NORDKLIM project plans by DRIFT v.1.1."*, Oslo, April 3rd, 1999.
- [6] *DNMI-Report no. 31/99 KLIBAS "Monitoring KLIBAS system mail by MAILSTAT v.2.0."*, Oslo, April 14th, 1999.
- [7] *DNMI-Report no. 12/99 KLIMA "Quality Control of Climate Data: An Automatic Approach"*, Oslo, March 19th, 1999.

Adding linear interpolation of air temperature to the ALA2TELE data transfer program

Petter Øgland

Norwegian Meteorological Institute
Niels Henrik Abels vei 40
P.O. Box 43, Blindern
N-0313 Oslo, NORWAY
(subm. April 15, 1999)

ABSTRACT

The purpose of the program ALA2TELE is to convert observations from the datatable ALA to a synop format and to insert these into the datatable TELE. This note addresses a problem with the program causing it to top the problem statistics in the KLIBAS maintenance system.

The ALA2TELE program

July 1997: The AUTO_INN system [1] was investigated and partly reprogrammed from scratch. One of the new features in version 1.1 was a daily quality evaluation called AUTO_MKK, producing the same type of results as the monthly orientated MKK.

February 1998: In order to prevent the SYNO_INN system [2] to break down because snr was not defined, a program ALA2TELE_PARA was constructed that inserted snr from auto init files in case the new station was an AWS.

November 1998: The version 1.1 of ALA2TELE_PARA [3] has been constructed due to two more or less problematic events during the last months. The first change was to have the program select the proper snr/stnr from ST_INFO when there are more snr to select from. The other change is to use one of the test station stnr when no snr is found in ST_INFO. Both problems caused serious problems for the SYNO_INN system during October and November 1998.

April 1999: In the version 1.1 of ALA2TELE [4] piecewise linear interpolation is used for missing hourly values of temperature from ALA when producing temperature for the TELE datatable. No updates are done in the ALA table however. In order to monitor the

ALA2TELE program in a better way, statistical process control have been added.

Each run by ALA2TELE is logged, registering program breakdown as a defect. The first log of the program is in September 1995. It was then run 30 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.

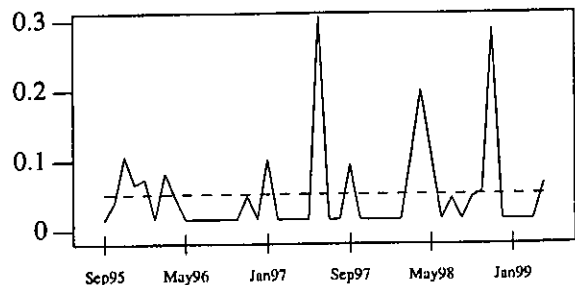


Figure 1. Relative number of defects

The average value dashed in the figure is 0.04. Statistics of this kind is present method for documenting general system quality and improvement.

The current problem

The message "Error: try spline interpolation for temperature in ALA for solving problems with ALA2TELE not being abl" was first recorded on an err file on Mon Feb 22 1999. We ask ourselves what can be learned about the KLIBAS system and ALA2TELE by understanding this problem, and how may it help achieving the ultimate goal of the system?

The ultimate goal of the KLIBAS system is to be totally self-contained in terms of automatically carrying out all operations of quality control of climate data with as little manual interaction as possible [5].

In order both to understand the problem and to solve it, a plan must be devised, an experiment must be carried out and the results must be interpreted.

Planning an experiment

The purpose of the program ALA2TELE is to convert observations from the datatable ALA to a synop format and to insert these into the datatable TELE.

The figure shows the daily total of observations inserted from ALA into TELE. The plot is a part of the Statistical Process Control (SPC) for the system and consists of a dashed line representing the average value of the day for the process and the upper limit of the process (process capability) given in terms of six times the standard deviation of the process.

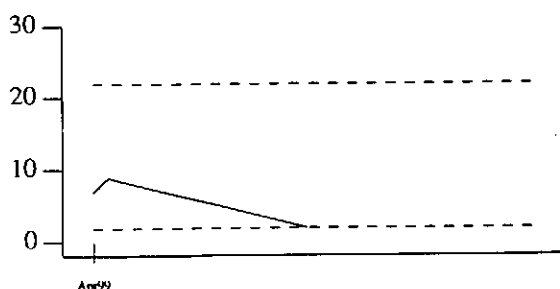


Figure 2. ALA2TELE evaluation by SPC

Running the experiment

According to specifications above, an experiment should be run and evaluated. The actual running of the experiment should be handled by the systematic running of the ALA2TELE program itself.

The ALA2TELE program is run automatically once a day as schedules by crontab vi that

AUTO_INN program. AUTO_INN also runs the AUTO_MKK program which makes use of the output from ALA2TELE. What the ALA2TELE program does is to read meteorological observations from the ALA data table, format these into synop format and insert the observations into the TELE and SYNOP2 data tables.

When the system has been running for sufficiently long time, it should again be evaluated, this note being a vital reference for documenting improvements.

Conclusion

As a step in solving the title problem in this note, an experiment has been designed, and measures on how to carry this experiment out have been described. The results will be evaluated when enough data is available for drawing conclusions.

As a plan for further development, it is needed to collect more statistics and find out if the SPC can capture the process in a better manner where not only the observations per se are counted but also each element in an observation.

However, in order to put the results in a wider context and prepare for the next set of experiments, literature in the field of research should be investigated in case there are methods and concepts that should be included before new experiments are designed.

References

- [1] DNMI-Report no. 59/97 KLIBAS "Innlesing AUTO_INN v.1.1 for AVS: Programmer mnd2ala, mnd2hla, ala2tele, adk og mkk.", Oslo, July 29th, 1997.
- [2] DNMI-Report no. 16/98 KLIBAS "Computer program ALA2TELE_PARA for automatic update of AWS in TELE_PARA.", Oslo, February 23rd, 1998.
- [3] DNMI-Report no. 63/98 KLIBAS "Inserting snr/stnr into TELE_PARA by ALA2TELE_PARA v. 1.1.", Oslo, November 13th, 1998.
- [4] DNMI-Report no. 32/99 KLIBAS "Adding linear interpolation of temperature in ALA2TELE v.1.1.", Oslo, April 15th, 1999.
- [5] DNMI-Report no. 12/99 KLIMA "Quality Control of Climate Data: An Automatic Approach", Oslo, March 19th, 1999.

Improving upon a solution for the PIO_INN data collection system

Petter Øgland

Norwegian Meteorological Institute
Niels Henrik Abels vei 40
P.O. Box 43, Blindern
N-0313 Oslo, NORWAY
(subm. April 21, 1999)

ABSTRACT

The PIO datatable is to contain observations collected by the "PC i observasjonstjenesten" routine. The program PIO_INN reads datafiles on monsoon:/usr/local/data (/opdata/automat/pio) into datatable PIO on DBSERVER. Statistics are provided with the program in order to make sure that observations are found and read. Summary statistics are checked daily and printed in the monthly KLIBAS statistics report. This note addresses a problem with the program causing it to top the problem statistics in the KLIBAS maintenance system.

The PIO_INN program

May 1998: PIO data ("PC i observasjonstjenesten") have been recorded on files since Mars 23rd 1998. The purpose of the program PIO_INN [1] is to automatically insert PIO observations into the PIO datatable in the KLIBAS database system on a daily basis.

July 1998: The report [2] gives a description of a revised version of the PIO_INN program, now reading data from a revised PIO format, inserting data into a differently designed PIO datatable and at the same time adding rows to the TELE table.

August 1998: The version 2.0 of PIO_INN [3] is significantly reprogrammed in order to handle data from the SAWS/mnd format in addition to the PIO/mnd format. The idea of the revised PIO_INN program is to store the manually observed part of the SAWS in the PIO table.

October 1998: The reason for publishing this version 2.1 of the PIO_INN program [4] is that the program is now significantly altered in order to read SAWS data and make sure that only the manual part of the observation sets are stored in the PIO table. In order to have better control over the PIO_INN system, new plots and statistics are added.

October 1998: In the version 2.1 of PIO_INN [5] it was realised that in order to collect only data rows containing real data, tests had to be applied in order to find out if that row was assumed to contain an observation or not. In each such case where a missing value was not clearly defined, and had to be assumed, the instance was logged. In the version 2.2 of PIO_INN this random number of errors is applied for defining whether the PIO_INN is in statistical control or not.

February 1999: A status report was written [6]. Personal computers have been used on observation sites since the autumn of 1997. At present there are 13 such PIO stations operating within the DNMI meteorological station network. Changes in the version 5 of the PIO software are now to be done. There may be capacity for another 20 PIO stations in the system before the end of 1999. The report contains administrative and economical analyses.

April 1999: Adaptions in were made in PIO_INN v.2.3 [7] in order to handle semi-automatic weather stations (SAWS) in an improved manner, changing the order to the columns containing weather information; V1, V2, ..., V7.

Each run by PIO_INN is logged, registering program breakdown as a defect. The first log of

the program is in May 1998. It was then run 152 times, failing 22 of these (14.5%). 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.

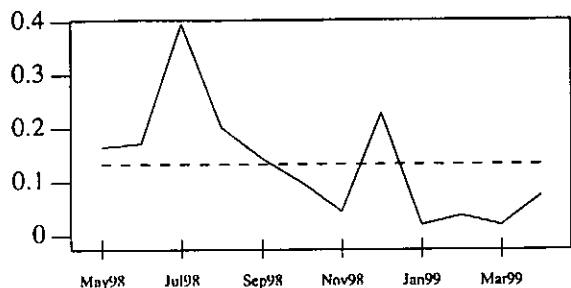


Figure 1. Relative number of defects

The average value dashed in the figure is 0.11. Statistics of this kind is present method for documenting general system quality and improvement.

The current problem

The message "Error: Aase reports problem with SAWS, V7 should be V6 etc. (note March 14th 1999) in KLIMA folder" was first recorded on an err file on Wed Apr 07 1999. On the folder a case of Fokstua (stnr 16610) is described where the station on February the 5th 1999 at 12:00 contained a value 5 in the column V7 and none in V6.

By investigating the source file that the PIO_INN computer program uses as input, we see that the weather information in V1, V2, ..., V7 are read from the columns _VT_old1, _VT_old2, _VT_new1 and _VT_new2.

The manipulation of these columns, in order to create the right values for V1, V2, ..., V7 is done in the function "testValues" in PIO_INN. In this function, the program assumes that v4, v5, V6 and V7 are stored in the columns _VT_old1 and _VT_old2. In particular, the column _VT_old1 is a concatenation of V7 and V6, while _VT_old2 contains V5 and V4.

In the computer program, however, this was mixed up, _VT_old1 assumed to contain V5 and V4 and _VT_old2 to contain V7 and V6.

Conclusion

The mix-up of columns _VT_old1, _VT_old2 on the data file and the columns V4, V5, V6 and V7 in the Oracle datatable PIO should be corrected. Automatic sorting of argu-

ments should included in the improved algorithm.

References

- [1] DNMI-Report no. 19/98 KLIBAS "Computer program PIO_INN.", Oslo, May 12th, 1998.
- [2] DNMI-Report no. 25/98 KLIBAS "PC i observasjonstjenesten: PIO_INN v.1.1.", Oslo, July 21st, 1998.
- [3] DNMI-Report no. 41/98 KLIBAS "PC i observasjonstjenesten: PIO_INN v.2.0.", Oslo, August 21st, 1998.
- [4] DNMI-Report no. 53/98 KLIBAS "PC i observasjonstjenesten: PIO_INN v.2.1.", Oslo, October 13th, 1998.
- [5] DNMI-Report no. 58/98 KLIBAS "PC i observasjonstjenesten: PIO_INN v.2.2.", Oslo, October 23rd, 1998.
- [6] DNMI-Report no. 07/99 KLIMA "PIO STASJONER. Status februar 1999.", Oslo, February 22nd, 1999.
- [7] DNMI-Report no. 34/99 KLIBAS "PC i observasjonstjenesten: PIO_INN v.2.3.", Oslo, April 21st, 1999.

Eliminating an ORACLE ERROR SQL-02112: "SELECT..INTO returns too many rows" from the SYNO_INN computer program

Petter Øgland

Norwegian Meteorological Institute
Niels Henrik Abels vei 40
P.O. Box 43, Blindern
N-0313 Oslo, NORWAY
(subm. April 22, 1999)

ABSTRACT

The purpose of the program SYNO_INN is to insert data from syno-files into datatables SYNOP, SYNOP2, TELE and TELE2. The program also generates a log and statistics for observations that are rejected for various reasons. This note addresses a problem with the program causing it to top the problem statistics in the KLIBAS maintenance system.

The SYNO_INN program

February 1994: In order to produce weekly up to date statistics for certain weather stations, fresh observations need to be inserted into the KLIBAS database. A program SYNO_INN [1] (FIFOfeed + FIFOread) has been implemented in order to read syno-files on SGI computer Tornado into Oracle datatable SYNOP every hour at 00:00, 01:00, ..., 23:00 UTC.

December 1994: In the version 1.1 of the SYNO_INN [2] program, bugs were removed and minor changes were made.

July 1995: The program SYNO_INN [3] was once again revised. Shell programming was used to a greater extent than what had been the case earlier on.

September 1995: A report [4] was written to document recent developments of the TELE/SYNOP system, including VNN program TROMSO and the SYNO_INN programs AUTO2TELE and GTS2SYN.

October 1996: SYNO_INN [5] was totally restructured and programmed from scratch. A quality control program SYNO_KONTR was created in order to compare values in the data tables SYNOP and SYNOP2.

January 1997: The program SYNO_INN [6] was extensively reprogrammed.

June 1997: The SYNO_INN program [7] was extended with routines to reduce the need for manual surveil of the program.

January 1998: The purpose of the revised version 3.3 of the computer program SYNO_INN [8] was to make it possible to insert chosen foreign synop observations into the TELE datatable. At the moment of its release, five test stations were being used.

October 1998: SYNO_INN [9] was revised due to an uncontrolled breakdown on Friday October 16th 1998. Errors in the code documented in report no. 03/98 KLIBAS were found and corrected.

October 1998: The 3.5 version of the SYNO_INN program [10] is augmented by statistical process control in terms of logging every format problem that arises when inserting into the Oracle database and logging every discrepancy having to do with observations not corresponding to definitions in TELE_PARA in order to use the log for statistical analysis.

November 1998: The prototype version of ALF2TELE updates only temperature from ALF to TELE. The program [11] is run from the SYNO_KONTR system, making sure that meter data are inserted into TELE before interpolation is executed. The program produces control charts

for daily statistical process control of the system.

November 1998: In the 3.6 version of the SYNO_INN system [12] the SPC statistics have been developed further. It turned out the program changes made for version 3.5 made the program critically slow, so in this new version reprogramming in order to make the routine more efficient have been applied.

January 1999: The version 3.7 of program [13] was made to prevent the system from breaking down in case of failure in the insert procedures INSERT_SYNOP, INSERT_SYNOP2 and INSERT_TELE. The adjustment was made to compensate on a problem of SYNO_INN breaking down on Sunday January 17 at 19:41 GMT as it was unable to extend the SYNOP-index. Still the program will not be able to extend the SYNOP-index, but it automatically generates an e-mail message to a person who may be able to do so and continues inserting data into the other data tables.

March 1999: The version 3.8 has been made in order to accommodate requirements for running the SYNO_INN program [14] on the vnn-y2k computer in order to test for Year 2000 problems. In order to achieve this, the program is now portable, and does no longer depend on files being stored on specific paths. The new version also reads data into the new TELE2 data table that is designed for simulating the TELE routine.

Each run by SYNO_INN is logged, registering program breakdown as a defect. The first log of the program is in May 1997. It was then run 212 times, failing 28 of these (13.2%). 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.

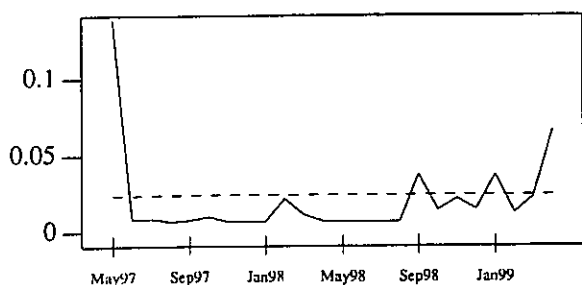


Figure 1. Relative number of defects

The average value dashed in the figure is 0.0174. Statistics of this kind is present method for documenting general system quality and

improvement.

The current problem

The message "ERROR: SQL-02112: SELECT..INTO returns too many rows" was first recorded on an err file on Thu Dec 31 1998. We ask ourselves what can be learned about the KLIBAS system and SYNO_INN by understanding this problem, and how may it help achieving the ultimate goal of the system?

This particular problem has occurred several times during the past five months. A superficial analysis was given in the KLIBAS research notes vol. 4 [15], but there is presently no actual understanding of why it breaks down in this manner.

The program breaks down in the procedure "statistics" when it tries to read observations from the SYNOP2 data table in to local variables. Apparently there is not enough space in the local variables in order to read the 36402 rows specified in SYNOP2, even though the local variables are defined to be able to contain 100000 values each.

The problem may have something to do with not enough memory space to allocate the right size.

Proposed solution

One way to handle the difficulty may be to read one station at a time instead of all stations as is done at the moment. This would also make it possible to remove the earlier ad-hoc solution consisting of not running the "statistics" functions on the last day of the month.

References

- [1] DNMI-Report no. 05/94 KLIBAS "Innlasting av synoptiske data til arbeidslager.", Oslo, February 22nd, 1994.
- [2] DNMI-Report no. 40/94 KLIBAS "Innlasting av synoptiske data til arbeidslager. Revidert utgave.", Oslo, December 20th, 1994.
- [3] DNMI-Report no. 18/95 KLIBAS "Innlasting av synoptiske data til arbeidslager. Versjon 2.0.", Oslo, July 4th, 1995.
- [4] DNMI-Report no. 22/95 KLIBAS "Programmer auto2tele, gts2syn og tromso i synoprutinen.", Oslo, September 14th, 1995.

- [5] *DNMI-Report no. 15/96 KLIBAS "Eksperimentell overføring av data fra syno-filer til Oracle-database. Versjon 3.0."*, Oslo, October 29th, 1996.
- [6] *DNMI-Report no. 04/97 KLIBAS "Overføring av data fra syno-filer til tabeller SYNOP og TELE. Versjon 3.1."*, Oslo, January 19th, 1997.
- [7] *DNMI-Report no. 55/97 KLIBAS "Dataoverføring SYNO_INN v.3.2 fra syno-filer til tabeller SYNOP og TELE med utvidet sikkerhet."*, Oslo, June 7th, 1997.
- [8] *DNMI-Report no. 03/98 KLIBAS "SYNO_INN v.3.3: Revised for inserting international synops into TELE."*, Oslo, January 14th, 1998.
- [9] *DNMI-Report no. 54/98 KLIBAS "Reading data from syno-files into KLIBAS: SYNO_INN v.3.4."*, Oslo, October 17th, 1998.
- [10] *DNMI-Report no. 59/98 KLIBAS "Reading data from syno-files into KLIBAS: SYNO_INN v.3.5."*, Oslo, October 27th, 1998.
- [11] *DNMI-Report no. 60/98 KLIBAS "Updating METAR from ALF in TELE by ALF2TELE."*, Oslo, November 3rd, 1998.
- [12] *DNMI-Report no. 65/98 KLIBAS "Reading data from syno-files into KLIBAS: SYNO_INN v.3.6."*, Oslo, November 21st, 1998.
- [13] *DNMI-Report no. 05/99 KLIBAS "Reading data from syno-files into KLIBAS: SYNO_INN v.3.7."*, Oslo, January 18th, 1999.
- [14] *DNMI-Report no. 20/99 KLIBAS "Reading data from syno-files into KLIBAS: SYNO_INN v.3.8."*, Oslo, March 23rd, 1999.
- [15] *DNMI-Report no. 13/99 KLIMA "KLIBAS research notes volume 4"*, Oslo, April 1st, 1999.

Preventing the SYNO_KONTR system from updating temperature where FLTT equals one

Petter Øgland

Norwegian Meteorological Institute
Niels Henrik Abels vei 40
P.O. Box 43, Blindern
N-0313 Oslo, NORWAY
(subm. April 23, 1999)

ABSTRACT

The purpose of SYNO_KONTR is to manage the interpolation and correction programs in the TELE/SYNOP system and thereby prepare the TELE datatable for monthly climatology statistics and for daily use at 8 or 9 UTC. The output from SYNO_KONTR contains information on observation quality and problems in the automatic quality control routine for quality processing TELE. This note addresses a problem with the program causing it to top the problem statistics in the KLIBAS maintenance system.

The SYNO_KONTR program

October 1996: SYNO_INN was totally restructured and programmed from scratch [1]. A quality control program SYNO_KONTR was created in order to compare values in the data tables SYNOP and SYNOP2.

August 1997: The SYNO_KONTR system [2] was remodelled and added to the TELE/SYNOP system as a monitor. The purpose of SYNO_KONTR was to start quality control and interpolation programs on a daily basis and generate statistics from the execution of these programs.

January 1999: The version 2.1 of SYNO_KONTR [3] is expanded by including a test for checking for "non-defined" observations in the datatable TELE. If values are inserted by the SYNO_INN program or the interpolation programs (INTERPOL2 or INTERPOL_P0) in columns that should not have been updated, this program takes notice and include these problems along with a count of missing values in curves and lists updated daily. The purpose of all this is aid to help that the TELE data table contain all needed obser vations, but no more.

Mars 1999: The S-T-F system [4] is used for autmatic and manual quality control of real-

time data at DNMI. The report describes the development and performance of the system, with emphasis on how the quality checks relate to WMO guidelines. The S-T-F program is part of the Quality Assurance System at DNMI, and statistical process control is used for monitoring the performacne and quality of the data processing of read-time data in order to find ways of improving the system.

Each run by SYNO_KONTR is logged, registering program breakdown as a defect. The first log of the program is in July 1997. It was then run 15 times, failing 6 of these (40.0%). 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.

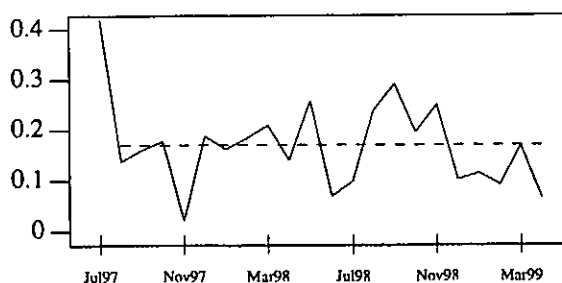


Figure 1. Relative number of defects

The average value dashed in the figure is 0.15. Statistics of this kind is present method for documenting general system quality and improvement.

The current problem

The message *"Error: Stein adds FL=1 for EIK HOVE, but everyday the obs is removed and the S-T-F list remains long"* was first recorded on an err file on Thu Apr 15 1999. We ask ourselves what can be learned about the KLIBAS system and SYNO_KONTR by understanding this problem, and how may it help achieving the ultimate goal of the system?

The ultimate goal of the KLIBAS system is to be totally self-contained in terms of automatically carrying out all operations of quality control of climate data with as little manual interaction as possible [5].

The major difficulty in this case is to find which program where the update of FLTT:=NULL is done. How this type of search could be carried out automatically is difficult to say. In the sections below, however, we will describe how the search is carried out manually, perhaps giving a sort of tentative algorithm for performing the same task automatically.

Searching the problem

As SYNO_KONTR is the program the is simulating and running all the other programs in case of automatical updates, the source code for SYNO_KONTR should be investigated first.

By using the formula `"grep -i FLTT syno_kontr.pc"` a few relevant lines from the `syno_kontr.pc` source file appears, but none that are connected to the actual problem. Obviously a more sophisticated form of search is needed, something that would allow us to search in sequences of directories for possible FLTT updates.

Using the `kapo` crontab as a prototype, a call of the type `"find `kapo -name '*.pc'` -exec grep -i FLTT= {} ; | grep -i update"` would perhaps be better. Alas it is not. Although all the relevant files are searched, there are no updates.

The next strategy will then be to search the programs called by SYNO_KONTR one by one, trying to understand what they do and find out if this may influence the FLTT flag in any manner.

Searching SYNO_TEST

In this program there is no reference to the FLTT column. A function `"update_n_tele"` is used for updating the N/FLN columns in TELE, and a similar function `"update_n_synop2"` is used for SYNOP2. Delete functions are used for preventing the SYNOP2 table from growing to large, and there is a delete function for removing columns in the TELE_PARA data table.

Searching MANGELLISTE

In this program there are specific references to the FLTT column, but there are no updates or deletes.

Searching INTERPOL1

In this program there is no reference to the FLTT column and no delete from the KLIBAS database. Updates are done on the TELE table, filling in data from SYNOP2, but the flag columns are not used.

Searching TELE_UFULL

The TELE_UFULL program is run by a system call, and we do not know its source code. The purpose of the program is to fill in empty rows, however, so it is not expected that this program has anything to do with updating the FLTT columns.

Searching SYNO_UFULL

There is no reference to FLTT here. There are no delete functions, and there are not update functions.

Searching ALF2TELE

In this program there are explicit references to the FLTT column, and the program does in fact update TT/FLTT in TELE using METAR data from ALF. Each update, however, is accompanied with FLTT being set equal to '6', so this program cannot either be the one causing problems. There are no delete functions.

Searching INTERPOL2

Here we have two references to the FLTT column, but they both belong the function that updates FLTT with '4' when inserting statistical interpolations.

Searching CONTROL1

In this program there is no reference to the FLTT column. There is no SQL delete sentences either, and no update sentences.

Searching CONTROL2

In this program the TT/FLTT columns are updated, but FLTT is assigned the value '4'. There are no delete sentences.

Searching INTERPOL3

Here there is no reference to the FLTT column. All the update sentences relate to RR/FLRR, TN/FLTN and TX/FLTX. There are no delete sentences.

Searching S-T-F

This program contains several references to the FLTT column, and it also contains update sequences, but what these sentences describe are updates where the value of TT is removed and the value of FLTT is set like '5'. There are no delete sentences.

Searching CHECK_H_STAT

The column FLTT is not used by this program. Here are no deletes and no updates.

Searching CHECK_STATUT

Here there is no reference to the FLTT column, and no update or delete sentences either.

Searching FLAG_STAT

Here there is a reference to the FLTT column, but no updates or deletes.

Searching SYNO_ESPEN

No reference to the FLTT column in this program, and no references to delete or update either.

Searching ALA2ALV

The program ALA2ALV is not explicitly referred to in the SYNO_KONTR system code, but as no problems are found in the relevant programs, this programs should be checked as well.

Here we have references to the FLTT column, and the TT/FLTT columns are updated with FLTT assigned the value '5' in TELE and '1' in ALV.

A new approach

As neither the search via grep or manual check of each of the programs one by one solved the problem, another type of approach have to be taken. By running the SYNO_KONTR program we know that the FLTT columns are being updated. This is a fact, but we do not know how it happens.

What can be done, although it takes some time, 43 minutes on the average, is to run the complete SYNO_KONTR program, and check status in the TELE data table at various instance in order to find out which particular program is causing the problem in this manner.

This is not a perfectly easy way of doing it as some of the subprograms run very quickly and others take considerable time. At this point, however, it seems like the most reasonable thing to do.

First we check that the TELE table contains values '1' in the FLTT column for Eik Hove (stnr 43010), and then we start the system, which immediately starts SYNO_TEST. ALF2TELE is also started, and the two programs run in parallel.

When the INTERPOL2 program is running, there has still not been any updates on FLTT, but from the INTERPOL2 program, other subprograms are run, such as INTERPOL_PO, INTERPOL_TT and HIRLAM.

At a time when the HIRLAM program is running, suddenly the flags in FLTT have been removed. The HIRLAM program does not contain delete sentences, but it does contain sentences for updating TT/FLTT, although it will only do so for stations that are defined in the MIOP data table.

More than that, when checking the output of the program, it is seen that it does indeed execute an update for 43010.

```
UPDATE TELE x set (x.TT,x.fTT) = (select
(8.0*y.TT_K)/100.0,'4'          from      veri-
fop.hirlam10_1425@miopdb y where y.aar=x.aar
and y.mnd=x.mnd and y.dag=x.dag and
y.tim=x.tim and y.prog=6 and ((y.aar=4 and
y.mnd=1999) or (y.aar=4 and y.mnd=1999)))
where x.stnr in (select z.stnr from TELE_PARA z
where z.TT=1 and z.stnr=43010) and
x.stnr=43010 and x.TT is null;
```

In the report generated by this program, it states: 42 updates for TELE (TT) where stnr=43010, which is exactly the number of unfortunate updates found in the TELE table. In other words, we have found the solution to the

problem being the updates done by the HIRLAM program, trying to insert values into TELE without regarding the information on the FLTT column.

The solution to the problem also explains why the grep approach didn't work. It seems rather difficult to locate and correct these types of problems automatically.

References

- [1] *DNMI-Report no. 15/96 KLIBAS "Eksperimentell overføring av data fra syno-filer til Oracle-database. Versjon 3.0."*, Oslo, October 29th, 1996.
- [2] *DNMI-Report no. 60/97 KLIBAS "Døgnlig driftskontroll for SYNOP: syno_kontr v.2.0."*, Oslo, August 2nd, 1997.
- [3] *DNMI-Report no. 06/99 KLIBAS "SYNO_KONTR v.2.1: Daily interpolation and quality control."*, Oslo, January 20th, 1999.
- [4] *DNMI-Report no. 11/99 KLIMA "Quality Control of Meteorological Real-Time Data: Results from the S-T-F system."*, Oslo, Mars 13th, 1999.
- [5] *DNMI-Report no. 12/99 KLIMA "Quality Control of Climate Data: An Automatic Approach"*, Oslo, March 19th, 1999.

Solving a problem in the ALA2TELE_PARA program

Petter Øgland

Norwegian Meteorological Institute
Niels Henrik Abels vei 40
P.O. Box 43, Blindern
N-0313 Oslo, NORWAY
(subm. April 28, 1999)

ABSTRACT

When the SYNO_INN program finds snr in the syno-files not defined in the TELE_PARA table, a system call is made to the ALA2TELE_PARA program which then should select and insert such a snr. The program ALA2TELE_PARA will first search for possible snr/stnr among the AWS ini-files. If none are found, the program will search ST_INFO for stations that are operative. If this also fails the program will select a test station from within the range 99986 to 99999. This note addresses a problem with the program causing it to top the problem statistics in the KLIBAS maintenance system.

The ALA2TELE_PARA program

February 1998: In order to prevent the SYNO_INN system to break down because snr was not defined, a program ALA2TELE_PARA [1] was constructed that inserted snr from auto init files in case the new station was an AWS.

November 1998: The version 1.1 of ALA2TELE_PARA [2] has been constructed due to two more or less problematic events during the last months. The first change was to have the program select the proper snr/stnr from ST_INFO when there are more snr to select from. The other change is to use one of the test station stnr when no snr is found in ST_INFO. Both problems caused serious problems for the SYNO_INN system during October and November 1998.

April 1999: As a failure in the 1.1 version of ALA2TELE_PARA caused major breakdown of the complete SYNO_INN system on April 27-28 1999, the program has now been updated in order to prevent similar types of breakdowns [3]. Additional statistics for having the program looking after itself has also been added in this new version.

Each run by ALA2TELE_PARA is logged, registering program breakdown as a defect. The first log of the program is in February 1998. It was then run 49 times, failing 24 of these

(49.0%). 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.

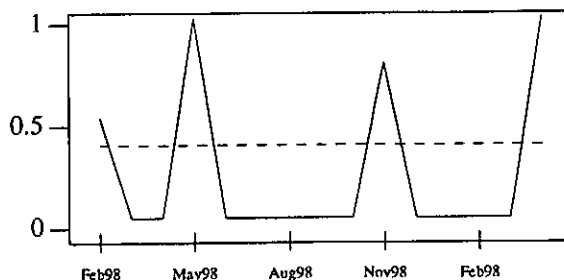


Figure 1. Relative number of defects

The average value dashed in the figure is 0.36. Statistics of this kind is present method for documenting general system quality and improvement.

The current problem

The error message reported by the program when running it with input '1999 04 01002', was that *no stations are registered in ST_TYPE WHERE stnr=99927 AND TYPE='A'*.

At this stage the program had already search for station no. synop=02 in zone 1 among

the aws files and found the file containing data from Verlegenhuken. On this file it said that the climate identification number for the station was 99927, but, apparently, when confirming the information against ST_TYPE, no station 99927 was found here.

Suggested solution

As it is not possible to update the ST_INFO table automatically, a person who has access to this table should be automatically e-mailed in order to prevent the system from breaking down again.

The ALA2TELE_PARA program should be augmented with functions in order to handle this.

It is also important that the complete SYNO_INN system does not collapse due to difficulties with ALA2TELE_PARA. Instead making SYNO_INN abort and terminate in error when there is problems with ALA2TELE_PARA, the program should only produce a warning and carry on as usual.

References

- [1] *DNMI-Report no. 16/98 KLIBAS "Computer program ALA2TELE_PARA for automatic update of AWS in TELE_PARA."*, Oslo, February 23rd, 1998.
- [2] *DNMI-Report no. 63/98 KLIBAS "Inserting snr/stnr into TELE_PARA by ALA2TELE_PARA v. 1.1."*, Oslo, November 13th, 1998.
- [3] *DNMI-Report no. 38/99 KLIBAS "Inserting snr/stnr into TELE_PARA by ALA2TELE_PARA v.1.2."*, Oslo, April 28th, 1999.
- [4] *DNMI-Report no. 12/99 KLIMA "Quality Control of Climate Data: An Automatic Approach"*, Oslo, March 19th, 1999.