Norwegian Meteorological Institute met.no

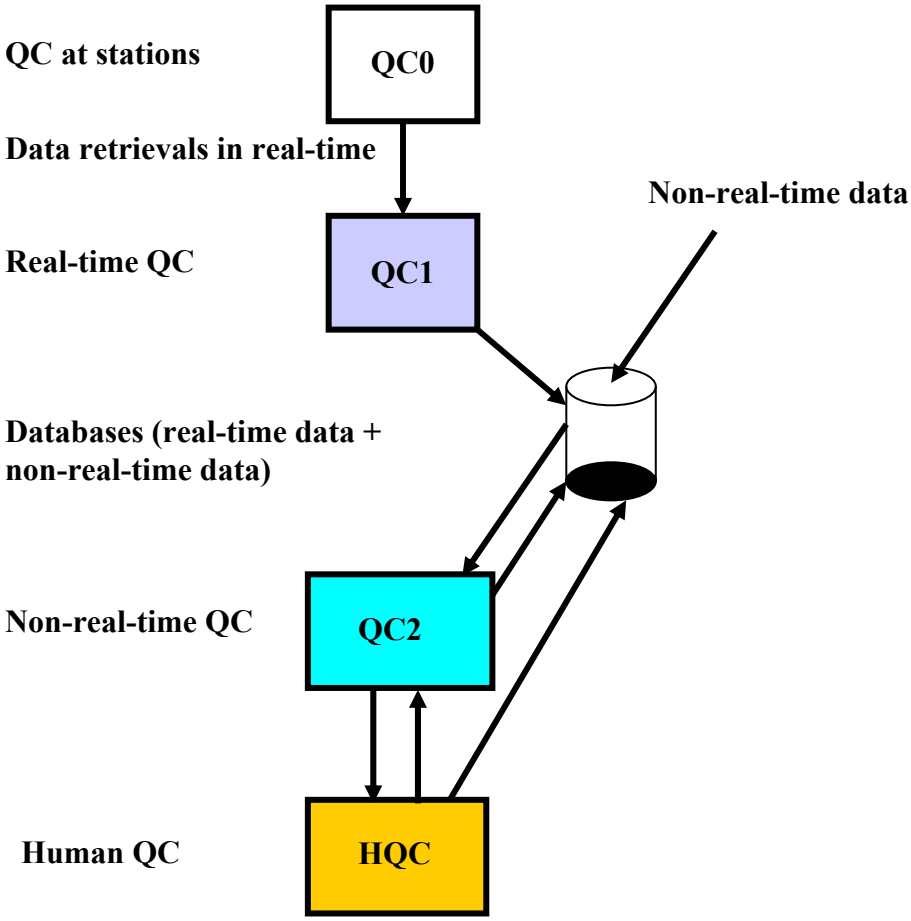
met.no report

no. 10/2004
Climate

Quality Control of Meteorological Observations

Description of potential HQC systems

Peter Svensson (ed), Halldor Björnsson, Antti Samuli,
Lars Andresen, Lisbeth Bergholt, Ole Einar Tveito,
Solfrid Agersten, Ola Pettersson, Flemming Vejen



DMI SMHI

NORDKLIM – Nordic co-operation within climate activities

Norwegian Meteorological Institute
met.no

report

Title Quality control of meteorological observations Description of potential HQC systems	**Date** 22.03.2004
Section Section for climate data	**Report no.** 10/2004
Author(s) Peter Svensson * (ed), Halldór Björnsson ¤, Antti Samuli §, Lars Andresen #, Lisbeth Bergholt #, Ole Einar Tveito #, Solfrid Agersten #, Ola Pettersson *, Flemming Vejen £ £ DMI, Denmark, § FMI, Finland, ¤ VI, Iceland, # met.no, Norway, * SMHI, Sweden	**Classification** ☒ Free ☐ Restricted **ISSN** 1503-8025 **e-ISSN** <nummer>
Client(s) NORDKLIM/NORDMET on behalf of the National Meteorological Services in Denmark (DMI), Finland (FMI), Iceland (VI), Norway (met.no) and Sweden (SMHI)	**Client's reference** <referanse>

Abstract

This report is prepared under Task 1, in co-operation with Task 2, in the Nordic NORDKLIM project: Nordic Co-Operation within Climate Activities. The NORDKLIM project is a part of the formalised collaboration between the NORDic METeorological institutes, NORDMET.

The report focuses on describing different presentation systems with potentials for human quality control (HQC). One system is a commercial HQC software product. Others are based on forecast presentation systems, which have potentials to be expanded with HQC parts. There is also a solution to develop a system from scratch. This document describes the present situation at the end of the year 2003, and gives a more technical view about potential HQC systems. It also gives a proposal of how to proceed with co-operation on technical solutions around HQC.

Keywords

Quality control, Meteorological observations, HQC systems, Technical information, NORDKLIM

Disiplinary signature	Responsible signature
_____	_____

Postal address P.O.Box 43, Blindern NO-0313 OSLO Norway
Office Niels Henrik Abelsvei 40
Telephone +47 22 96 30 00
Telefax +47 22 96 30 50
e-mail: met@met.no **Internet:** met.no
Bank account 7694 05 00628
Swift code DNBANOKK

Foreword

This report is prepared under Task 1, in co-operation with Task 2, in the Nordic NORDKLIM project: *Nordic Co-Operation within Climate Activities.* The NORDKLIM project is a part of the formalised collaboration between the NORDic METeorological institutes, NORDMET.

The main objectives of NORDKLIM are:
1). Strengthening the Nordic climate competence for coping with increased national and international competition
*2). Improving the cost-efficiency of the Nordic meteorological services (i.e. by i*mproving procedures for standardized quality control & more rational production of standard climate statistics)
3). Coordinating joint Nordic activities on climate analyses and studies on long-term climate variations

The NORDKLIM project has two main tasks:

1. Climate data (Network design, Quality control, Operational precipitation correction, long-term datasets).
2. Climate Applications (Time series analysis, use of GIS within climate applications, mesoscale climatological analysis).

A detailed description of the project is given by Førland et al. (1998).

NORDKLIM is coordinated by an Advisory Committee, headed by an Activity Manager. Each of the main tasks is headed by a Task manager.

The Advisory Committee in NORDKLIM is presently consisting of:

Hans Alexandersson, SMHI
Eirik J. Førland, DNMI (**Activity Manager**)
Raino Heino, FMI
Trausti Jónsson, VI
Lillian Wester Andersen, DMI

The present task managers are: Task 1: Caje Jacobsson (SMHI), Task 2: Ole Einar Tveito (met.no).

The addresses of the Nordic Meteorological Institutes are:
Denmark: Danish Meteorological Institute, Lyngbyvej 100, DK-2100 Copenhagen, Denmark, (www.dmi.dk)
Finland: Finnish Meteorological Institute, P.O.Box 503, FIN-00101 Helsinki, Finland, (www.fmi.fi)
Iceland: Veðurstofa Íslands, Bústaðavegur 9, IS-150 Reykjavik, Iceland, (www.vedur.is)
Norway: Norwegian Meteorological Institute; P.O.Box 43 Blindern, N-0313 Oslo, Norway, (http://met.no/)
Sweden: Swedish Meteorological and Hydrological Institute, S-60176 Norrköping, Sweden, (www.smhi.se)

Contents

Background

The Task 1 group in the Nordic NORDKLIM project works on issues concerning quality control of meteorological observations. One suggestion that arose within the group was to develop a common HQC (Human Quality Control) system in order to better allocate resources, to improve the quality of data and to make quality control more effective than it is at present. Currently, the Nordic countries use different QC-systems, but all need to construct a new interface to the data in the near future.

This report describes presentation systems with potentials for HQC. FMI in Finland and VI in Iceland have already started to discuss techniques for a new HQC-system, and done some preliminary tests. The Norwegian meteorological institute is developing an application that runs on top of met.no's digital analysis systems, DIANA. SMHI in Sweden has collected information about an existing system, QualiMET, developed by consultants used by Deutcher Wetterdienst (DWD). DMI in Denmark is participating in the development of a new presentation system, Ninjo, in co-operation with DWD, Meteoswiss and the Meteorological Service of Canada.

These systems can be utilised as a starting point for a new common HQC platform. They represent different ways of solving the problem. One system is a commercial HQC software product. Others are based on forecast presentation systems, which have potentials to be expanded to include HQC. Finally, one possibility is to develop a new system from scratch. In a common approach, regardless of which system will be utilised, each country has to make changes to access their existing systems and databases.

This document describes the situation at the end of the year 2003, and gives a technical overview of potential HQC systems. As technology changes, this description may become inaccurate within a few years. However, for the purposes of this compilation, a questionnaire was sent out to the participating members to get info on each system and to make it possible to compare the systems.

This report concludes with a suggestion that rather than aim for a common system, we could aim for harmonizing QC controls, and make better use of resources by the sharing of algorithms and modules. This is treated in further detail in the concluding section.

1. HQC Nordic

Authors: Halldor Björnsson VÍ and Antti Samuli FMI

1.1. Introduction

Initially this system was referred to as “a poor man’s HQC system”, but it got renamed to HQC Nordic at NORDKLIM meeting no 12 in 2003. The term “poor man’s version” refers to the fact that the idea is to develop various simple HQC programs (or components) that can interact, - using national weather services own staff rather than buying a complete package (or a “solution” in the present day jargon) from an external provider. The idea is that we could leverage work done in several locations into one package that allows for basic HQC.

The “poor mans” tag on the name simply reflects that costs in such a system are different from the costs associated with an externally purchased solution. In the poor man’s HQC costs accumulate over a period of time, while the system is in development, - however, the system can be used before all components are in place. The development period may become quite extended, although the simplest parts of the system could be implemented in the early stages of development. Perhaps this is the best argument for such a system, this way users of the system have much more of say in what goes into it. Often externally purchased solutions will be cumbersome in tasks that users need to perform frequently, but on the other hand have well optimised ways of doing tasks that users rarely, if ever, need done.

1.2. The basic structure of the system

The current version of this idea centres on the notion of a three-tier architecture. In such a system there are three main actors, the client, the server and the database (Figure 1). The end user or the client runs a web browser, which talks to the web server that receives a request for a web page from the user, and the database that receives queries from the server.

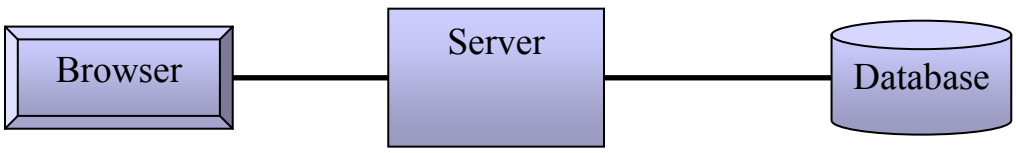


Figure 1. A three-tier architecture

This architecture has been compared to the division of labour in an old manor house. The master asks the butler to get a light meal. The butler goes down to the kitchen and has the maid collect the right ingredients. The butler then arranges the food items on a plate, takes it back up and serves it to the master.

The thing with this analogy is that just as this structure can survive an upgrade of the butler, maid or even master, the three tier architecture is not particularly dependent on which browser (explorer/netscape/mozilla/opera) is being used, what web server is being used (apache is by far most common, but there are alternatives) or what database the institution uses (oracle/DB2/Postgres/etc). This flexibility can be achieved as long as the server can run certain basic programs (java/perl/etc). The web pages requested by the client are dynamically generated, that is for the most part they do not exist prior the request being sent to the server.

Since all Nordic NMSs have different database structures at the moment, this approach can help to use same components while handling data in a database. Web server is the key part. Database related functions should be separated from web client serving functions. This allows using common client-serving programs while actual data come from different database for each country. Only database related functions need to be re-written for each country. Indeed, there are available libraries (such as the proprietary RogueWave Tools) that make it possible to write database related functions in a platform independent manner, with the library handling the translation of the function for each specific platform. If such a system was to be employed, even less work might need to be expended for each country. It should be noted, however, that regardless of system there are differences in table definition from one country to another, and hence the database access routines will always have to be tailored in each country.

Many database vendors will sell a three-tier web solution; both IBM/DB2 and Oracle come with an "application server" which helps implementing the above solutions. Currently the most common platform for these services is J2EE (Java 2 Enterprise Edition), which vendors generally supply.

1.3. Using the system

The following section provides our idea of how this system might be used, and what the simplest version of the system might entail.

The user starts by going to a web page with quality control results for a specific time, selected by the user. In what follows we will assume these are presented as coloured dots on a map, the colour code reflecting the "grade" obtained by the observation. The map might just as easily show observations for a given time.

The user can click on a point that is of interest, which brings up another web page. This webpage which we will call the "station page" will offer the user a selection of links. These links will include such things as "station description", "summary of statistics", and links to examine observations from the station in more detail. The observation links would show the observation at any given time. Furthermore there would be a link to see a timeseries of a specific parameter.

One way of making the timeseries is to use the scalable vector graphics (SVG) language, which allows SVG-enabled browsers to plot timeseries, and indeed most other vector graphs. SVG is similar in scope to Macromedia's Flash. It offers anti-aliased rendering, pattern and gradient fills, sophisticated filter-effects, clipping to arbitrary paths, text and

animations. It differs from Flash, in that the W3 consortium recommends it and that it is XML-based as opposed to a closed binary format. Currently the biggest obstacle to SVG is the fact that not all browsers support it. An alternative to SVG is to do the plotting within a Java applet. Java comes with plotting libraries that greatly facilitate the task of plotting timeseries. Tests at the VI have shown this to be a viable solution.

If the user decides that the observation is fine, the windows can simply be closed. If the user decides that the observation needs correcting, a link on the station page invokes an edit tool. Tools to edit database tables are usually sold with databases, so this is not likely to be a difficult task. However, if we use such specific tools, it is likely that the institutions would need different implementations.

As the different institutions have different databases, the construction of a common editing tool would be problematic. Such a tool would have to rely on html-forms or some other database independent ways and means, and might become impractical, especially if ready-to-run editing tools are already available from the database vendor.

It would also be of interest to have an automatic suggestion for the editing step. The details of how the suggestion is calculated are not relevant here, but would most likely be a natural extension of procedures used in QC1 and QC2. However, assuming we have implemented common methods for obtaining suggestions, the results from these could be accessed through a link from the station page.

1.4. Implementation

This document briefly describes an HQC system that is modular, and web-based. The user accesses it simply as webpages, and these pages are generated automatically from the user selection. The three-tier architecture means that the routines that generate the webpages are general enough to be used in all countries. A system like this can be built slowly, with new modules being added as they get developed, and at the users pace. Some of the proposed modules do exist already. At the VI there exists software to generate a map as a gif image and place it in a webpage with “hot-spots”, i.e. with points on the map that serve as links to other webpages. The map itself is generated using the GMT package, but the tools that generate the webpage and place hotspots and links on the page are standard Unix shell scripts. The latter are not dependent on the map generator, and some other software could generate the image. The FMI has uses Java and pre-generated maps for similar purposes.

To turn the present routines into an HQC program, a tool to plot time series is needed, as are rudimentary editing capabilities. The best way to proceed with a program like this is to start with the simplest possible system and then build on that by developing further modules. Thus we envision that a tool with plotting and editing capabilities would first be set-up in all countries, and then users would decide which extra tools would be most useful.

The chief benefits of the above approach are the following:
Low start-up cost. We start using the system early on when only a few modules are ready.

Knowledge retained locally. As most of the software is developed by institution staff any knowledge acquired en-route will reside with the institutions themselves.
Flexibility. The system is highly configurable.

As many of the components do already exist, it seems likely that a system such as this could be implemented in the space of a month or two, if full-time resources are committed to it. However, such an implementation is would be limited in scope, and the addition of further capabilities would take more time.

Chapter 2 gives an overview of the HQC system that is under development at the Norwegian meteorological institute. Chapter 3-4 show other solutions or possibilities concerning HQC systems, more or less based on commercial platforms.

2. KVALOBS-HQC

Authors: Lars Andresen, Lisbeth Bergholt, Ole Einar Tveito and Solfrid Agersten, met.no

2.1. Overview

The HQC-system is mainly developed within the ongoing met.no KVALOBS project. In addition GIS-tools are available through applications developed by the Climatology Division.

2.1.1. KVALOBS-HQC

The program returns different lists of observations, where the values can be corrected. The program communicates both with the database and with a visualisation program (DIgital weather ANAlysis, DIANA), where the observations, original or corrected, can be shown on a map together with fields, radar images, satellite pictures, time series presentation, other observations, etc. See more details in Andresen et al. (2003a).

In the manual control routine at met.no the Diana tool is in use for visualisation only. The map has no connection to the KVALOBS database yet.

2.1.2. GIS

The GIS-tool is an application using ESRI ArcIMS, which is an Internet Map Server program, as platform. In the application a topographical map with a convenient visualisation of the terrain and with all weather station locations is presented. Additional geographical data layers (rivers, lakes, land use, infrastructure etc) can easily be added in the view.

Shapefiles (ESRI, 1998) containing the location of weather stations are weekly updated. Observation can be selected from the database, and joined into the shape file within the ArcIMS application. There will also be developed a tool for presenting time series of an element at a given station.

The GIS application programme is under development.

2.2. Hardware

The two systems can execute on the following platforms:

- KVALOBS-HQC
  Linux, minimum 512 MHz CPU, 256 MB Ram (recommended 1 GHz, 512 MB)
  Special programs: Qt 3.1.2, Mesa 5.0, Gcc 3.2.2, but it will probably work with older versions too.

Compilers: C++ -API

- GIS
  Special programs: ArcIms from ESRI
  (http://www.esri.com/software/arcims/index.html)
  Server: Linux/Windows/Sun … 500 MHz and 256 MB RAM.
  See ArcIMS System requirements:
  (http://arconline.esri.com/arconline/sysreqs.cfm?PID=6)
  Development: 128 MB RAM
  Client: Minimum 64MB RAM

2.3. Software

Some information on the planned quality control software products:

- KVALOBS-HQC
  Developed at met.no
  Owner of source code: met.no
  Source code: C++

- GIS
  Program platform, ESRI ArcIMS, commercial software.
  Owner of source code: ESRI
  Source code: Executable
  Application development at met.no (built-in program):
  Owner of source code: met.no
  Source code: JAVA/JavaScript
  Server:WEB-server and JAVA engine (i.e. Apache and Tomcat) and ArcIMS server.
  Development: ArcIMS management. Restrictions in licence: At present 3 developer licenses (only 3 users simultaneously)
  Client: Web-browser (Internet Explorer).
  Java plugin JRE 1.3.02 or higher

2.4. Connections

Some technical information on the database/file systems:

- KVALOBS-HQC
  Postgres database with C-API
  The system can call QC1/QC2 tests

- GIS
  Oracle database, connected by JDBC. Files: Shape, Coverage, TIFF
  The system cannot call QC1/QC2 tests

2.5. Cost

It is not easy to make an estimation of the cost, but we assume the following timetable for finishing the systems:

- KVALOBS-HQC
  Version 1.0: Autumn 2004

- GIS
  Version 1.0: A couple of months are needed + licence costs (~28.000 NOK/year)

3. QualiMET, Quality control and monitoring system for meteorological data

Author: Ola Pettersson, SMHI

3.1. Overview

QualiMET is a complete validation and monitoring system for meteorological data. The information is accessed from a database, subjected to various validation procedures and written back into the database as validated. The operator controls the procedure through a graphical user interface. The validation rules can be adapted according to requirements. The Deutcher Wetterdienst (DWD) is using QualiMET as from 2002.

3.2. Hardware

The application has 3-layer system architecture, schematically shown in Figure 2:

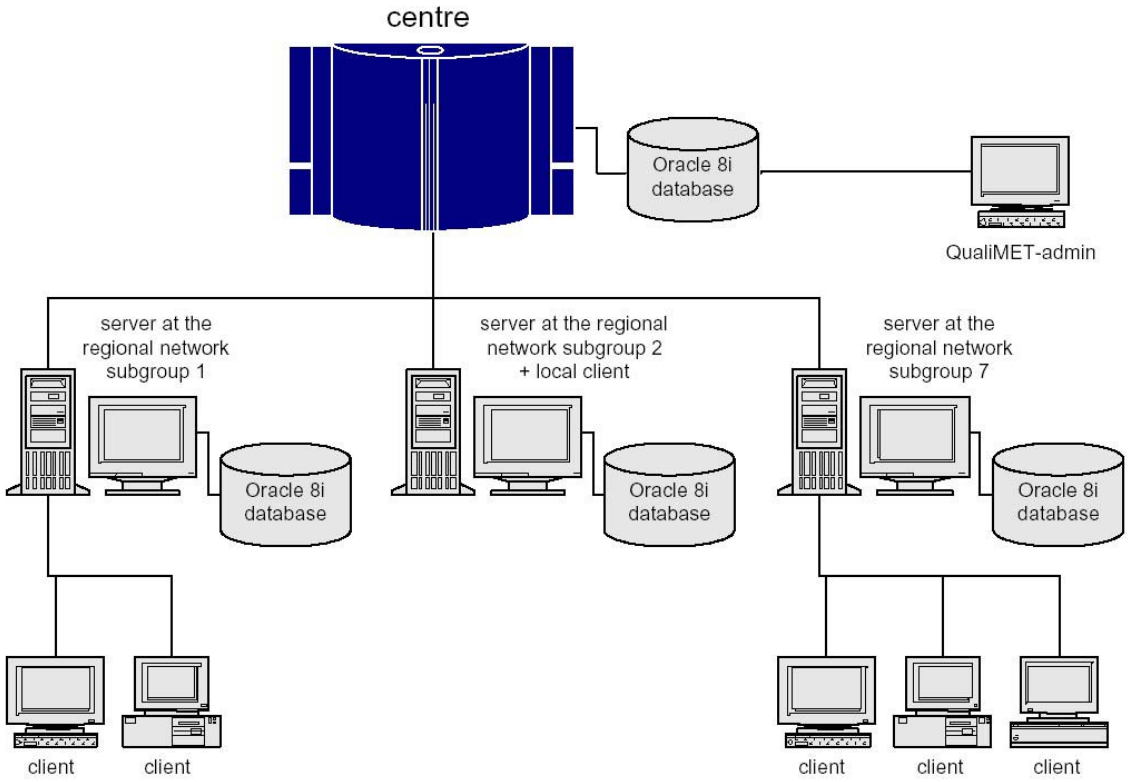


Figure 2. Overview of QualiMET's 3-layer system architecture consisting of 1:Main data base, 2: local server and 3: client.

Layer 1 is the main database with raw measurements, metadata and validation rules.

Layer 2 is the server, which receives the requested datasets from the main database and stores these in a local Oracle database for 14 days. For performance reasons, exchange with the main database is by PL/SQL procedures.

Layer 3 is the client, which visualises the data and allows interaction with the operator. The operator on one of the clients initiates validation, which runs on the layer 2 server. Client-server communication is by CORBA objects.

3.3. Platforms

Layer 1: may be any relational database
Layer 2: one, or several depending on number of controlled stations, 2- or 4-processor machine with 2-4 Gbyte RAM (Unix or Linux)
Layer 3: OS-independent javaprogram

3.4. Software

Developer: Ernst Basler+Partner GmbH, Potsdam, DE
Owner of the source code: DWD
Type of source code: Java
Restrictions in license: additional software development by original developer.

3.5. Connections

The data can be stored in any relation database. The system is connected to the database with PL/SQL or other solutions. The system cannot call QC1/QC2 tests, but these are implemented in the validation rules of QualiMET. The validation rules are stored in the database and can be adapted in each case to the current requirements. New rules may be composed and added by using QualiMET Admin which will transcribe the validation rule into SQL-statements that are accessed via CORBA objects during validation.

3.6. Cost

License cost for QualiMET 1.0 is € 10 000/year for at least 5 years. QualiMET admin cost € 2 500/year. The system is on the market today.

4. NINJO presentation system

Author: Flemming Vejen, DMI

4.1. General description

Ninjo is a new presentation system for meteorological observations and products (based on Java). It is developed in co-operation between Deutcher Wetterdienst (DWD, main responsible partner), Meteoswiss, Meteorological Service of Canada and Danish Meteorological Institute (DMI).

The system is under development and planned to be operational by the end of 2004. At DMI, it will be utilised by personal that are mainly dealing with weather service products, forecasting, as well as quality control and visual inspection of meteorological observations.

Ninjo uses a graphic user interface giving a complex variety of opportunities for presentation, visualisation and combination of different data types. Synop observations can be visualised on a map together with prognosis fields, radar images, satellite pictures, time series presentation, other observations, etc. The system can present a topographical map with a convenient visualisation of the terrain and with mapping of all weather station locations. It is possible to initiate different presentation setups.

There are currently not any plans for development of HQC modules in the common and general version of Ninjo, but each of the participating meteorological services is allowed to develop individual functionality's as for example HQC in their local version of Ninjo.

As in any meteorological presentation system, simple visual quality checks can be done, e.g. by comparison of different datatypes. It is planned to include in Ninjo tools for the quality control and corrections of meteorological products, but not observations.

4.2. A brief description of some requirements and technical information

It is required to develop a special interface between Ninjo and the surrounding environment, databases etc. to take care of data flow and communication. Observations are stored in a local database system.

CORBA (Common Object Request Broker Architecture) is required for Ninjo to execute and makes it possible for different computer applications to communicate and to work together over networks independent of source code. Using the standard protocol IIOP, a CORBA-based program from any vendor, on almost any computer, operating system, programming language, and network, can interoperate with a CORBA-based program from the same or another vendor, on almost any other computer, operating system, programming language, and network. CORBA can handle large number of clients, at high hit rates, with high reliability, and client and object may be written in different programming languages.

Source codes are C++, Java and C, and CORBA must be Java and C++ compatible. CORBA programming is done in ORBacus that is a high-performance, scalable, affordable and fully CORBA-compliant ORB for a broad variety of applications.

The hardware and software requirements are:

The system is run on Linux or Windows platforms, at DMI it runs on Linux. Tests of Ninjo have been run on the following target hardware: an Intel Pentium 4, 2.4 GHz CPU and 1 Gbyte RAM, with NVIDIA GeForce 4 MX 460, 64MB graphics card, and OpenGL 1.3 (GLX) drivers (compiled into the kernel).

4.3. Cost

It is not easy to make an estimation of the cost, and there is no rough estimate of it. It may be possible to estimate it from version 1.0 development costs (in man-years). The plan is to release version 1.0 by the end of 2004. License cost for Ninjo is not decided yet.

4.4. Discussion

There are several relatively expensive requirements. CORBA is required for Ninjo to execute, and it may be necessary to invest in one or more working stations. An interface between Ninjo and the surrounding environment, databases etc. must be developed and certain resources for development and maintenance be put.

Ninjo is a big system. Inclusion of HQC modules is a new aspect in Ninjo and the present general version, as well as the local version at DMI, will not imply such utilities. There is a need to make it possible to present quality control flags coming from the operational on-line QC system at DMI.

5. Conclusions

5.1. Is it possible to create a common system?

Today all NORDKLIM countries have their own QC systems. They have different structured databases and their own specialised developing tools which they are familiar to. Replacing all of this with a new common system is not going to be a success unless the new system is a lot better than the existing systems. Given the limited resources at the Nordic NMSs this seems unrealistic. The national institutes will most likely continue to use their own system even if NORDKLIM develop a new common HQC system.

– Our conclusion is, not to build a common system replacing our existing programs. We should instead focus on harmonizing QC control. Then we can easily share methods and algorithms and use the limited resources more effectively.

5.2. What is the objective with co-operation on HQC?

The best way to use our limited resources is to build a few common modules, which can be integrated into existing systems. The result of this development need not necessarily be a program or a routine. It can be a description for how a special module will be constructed. The result could also be a solution of how to best visualise a time series, or perform a spatial control using remote sensing data (ex. pictures) or gridded data set.

For example, a suggestion from Norway is to concentrate on developing a module with spatial controls in GIS, based on ArcIMS. Not all institutes have access to ArcIMS and can not benefit from this module. The delivery from Norway to NORDKLIM co-operation would therefore be a model describing the structure with algorithms and the data flow used in this module, exemplified in for example ArcIMS. Then the other countries can either take this ready to use ArcIMS-module, or build their own modules using other tools, (f. ex. statistical subroutines) to realise the same effect without ArcIMS. This can also be used for other modules in HQC that are requested by the users. These modules can for example be visualisation, QC-algorithms, homogenisation or distribution of accumulated precipitation.

If two institutes have the same systems a module can be implemented directly. If not, the module needs modification before integration. As visualised in Figure 3, not all countries may need a specific module; some of them may co-operate to develop a special module, partly by programming and partly by describing control methods.

A good example of how this cooperation is used in practice is the report concerning flagging parameters for QC (Andresen et al., 2003b). Finland and Norway are already using part of these flagging criteria in their systems. Iceland will now include these flagging methods in their new QC-system. Sweden has possibility to integrate it when they build their new archive system, and Denmark is positive towards integrating the flagging system in a planned modernization of their archive system.

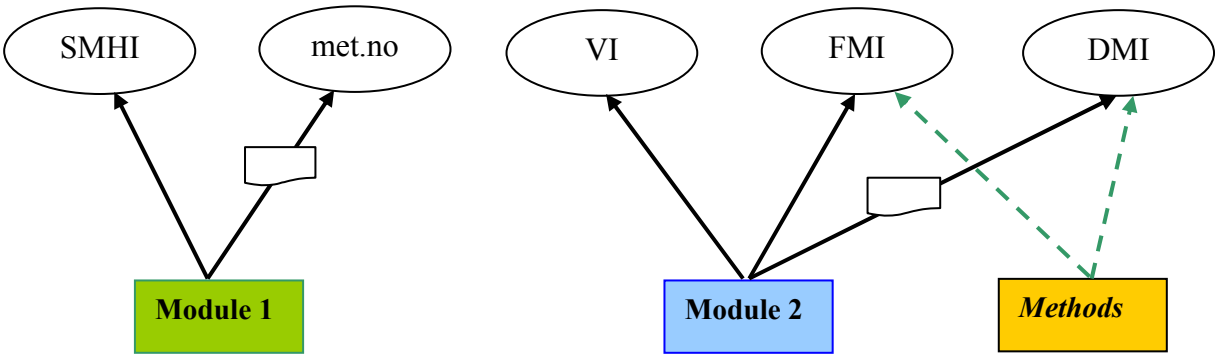


Figure 3. A concept where several institutes share one or more modules/methods with their own changes to fit respective systems.

– This example, of how to integrate new research, can always be used whenever an old HQC system needs to be upgraded. First, look for existing solutions for the problem. Make use of as much as possible of these, before starting to develop something new. If new functions have to be developed, share it with the other NORDKLIM members. Then they in their turn can benefit from these functions and spare resources.

In 2004 the working group will make a list of modules that are missing today and really needed in HQC. Many new ideas can be found in the report about recommendations for a common Nordic HQC system (Andresen et al., 2003a).

5.3. Future aspects

This report has concentrated on presenting technical issues about systems applied or under development at the individual Nordic NMSs. Apparently seeming to be quite different, they also have many similarities. They are based on a three-layer structure (Figure 1). This separation of data-storage, server processing and a client-side GUI makes it possible to develop implement joint algorithms in all these systems. If the systems are well designed, the different layers should easily be replaced, without too many adjustments in the remaining layers or modules.

The advantages and disadvantages by the presented systems are not emphasized in this report. Since a common Nordic HQC seems unrealistic, future developments should be focused on harmonizing HQC-algorithms. With this respect the performance and efficiency of the present applied systems should be considered. Weaknesses and strengths of the different platforms and data models should be identified. This is necessary in order to be able to choose the “best” algorithms or modules in any system. An approach of exchanging “core” (server-side) algorithms based on such principles should be an effective way to improve each of the systems in terms of HQC-usage.

A common Nordic HQC does not only depend on a joint technical solution. Along the national borders HQC-procedures depends on data from both sides of the border (or adjacent seas). The structures of the presented application should make it possible to apply the concept of distributed databases. This means that a server layer located anywhere

might select information from several databases. Such a distributed system is illustrated in Figure 4. This ensures access to the best available observations at any time, and also reduces problems with versioning and multi-storage.

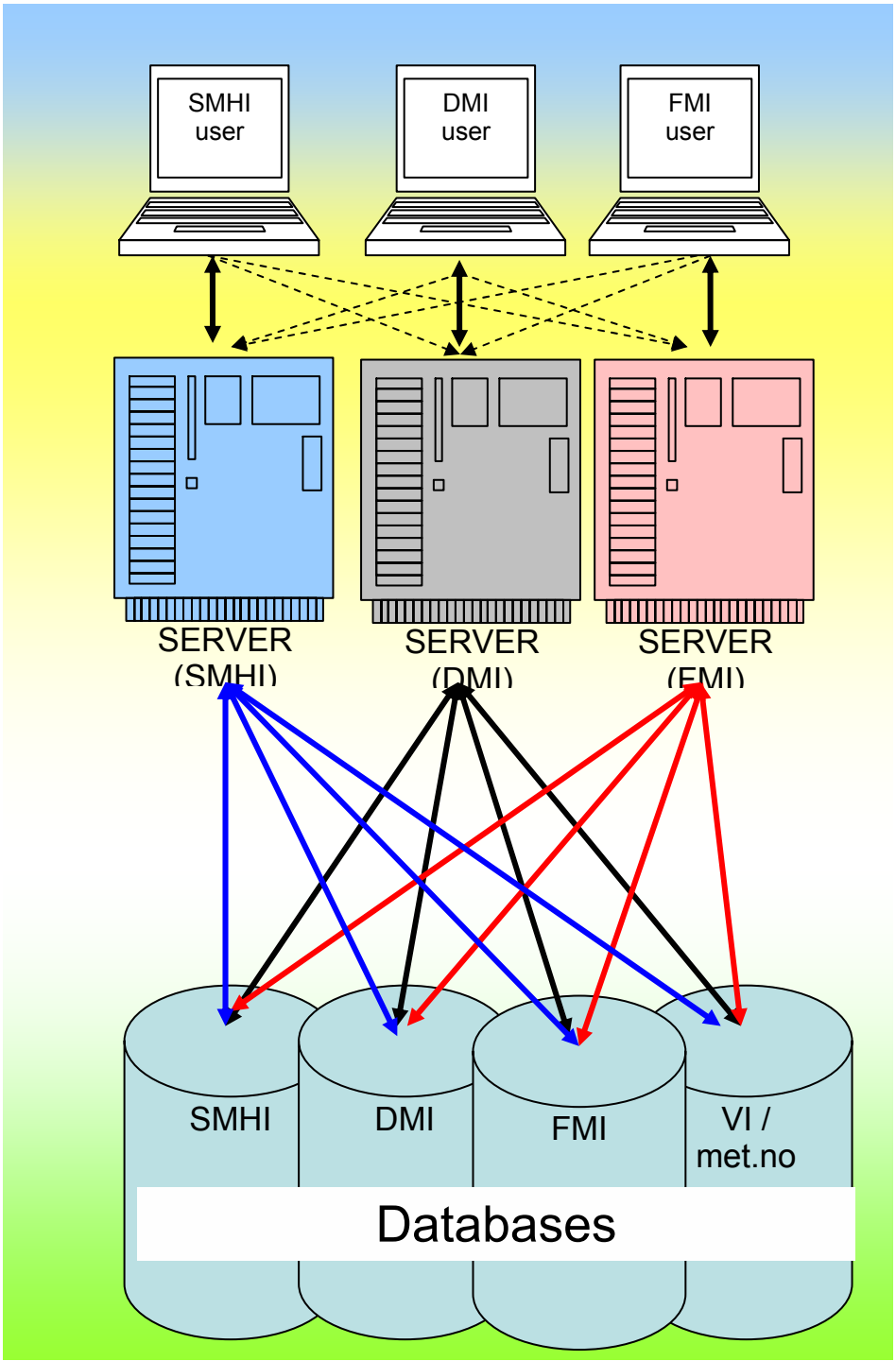


Figure 4. A distributed database approach to HQC-applications. Here applications at three institutes are connected to several databases. It is also technically possible to connect to other servers, as indicates by the dashed lines.

6. References

Andresen L., Björnsson H., Fredriksson U., Iden K., Jacobsson C., Pálsdóttir Þ., Pettersson O., Rissanen P., Samuli A., Vejen F., 2003a: Manual Quality Control of Meteorological Observations - Recommendations for a common Nordic HQC system.
met.no Report no. 9/2003 KLIMA, Norwegian Meteorological Institute, Oslo, 34 pp.

Andresen L., Björnsson H., Fredriksson U., Iden K., Jacobsson C., Pálsdóttir Þ., Pettersson O., Rissanen P., Samuli A., Vejen F., 2003b: Manual Quality Control of Meteorological Observations - Recommendations for a common Nordic end-user flagging system.
met.no Report no. 11/2003 KLIMA, Norwegian Meteorological Institute, Oslo, 13 pp.

ESRI, 1998: ESRI shapefile technical description, ESRI White Paper
(http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf)

Førland, E., Dahlström, B., Heino, R., Jónsson, T., Madsen, H., 1998: NORDKLIM (NORDisk KLIMasamarbeid) – Nordic Co-Operation within Climate Activities – Project description. Note, Oslo/ Norrköping/ Helsinki/ Reykjavik/ Copenhagen/ 08.06.1998, revised according to NORDMET rules in February 1999, 15pp.

Rissanen, P, Jacobsson, C., Madsen, H., Moe, M., Pálsdóttir Þ., Vejen, F., 2000: Nordic Methods for Quality Control of climate Data.
DNMI Report no. 10/2000 KLIMA, Norwegian Meteorological Institute, Oslo, 47 pp.

Vejen, F., Jacobsson, C., Fredriksson, U., Moe, M., Andresen, L., Hellsten, E., Rissanen, P., Pálsdóttir Þ., Arason,T., 2002: Quality Control of Meteorological Observations - Automatic Methods used in the Nordic countries.
met.no Report no. 8/2002 KLIMA, Norwegian Meteorological Institute, Oslo, 108 pp.