IMPLEMENTING AN ENHANCED AND INTEGRATED QUALITY ASSURANCE AND QUALITY CONTROL SYSTEM WITHIN THE MSC'S NEW DATA MANAGEMENT FRAMEWORK

L. Dale Boudreau*, and Alexander Zucconi
Meteorological Service of Canada, Downsview, ON, Canada

1. INTRODUCTION

The Meteorological Service of Canada (MSC) is in the second year of a multi-year project to overhaul its data management system for atmospheric, hydrometric and air quality monitoring data. Some aspects of the current system have been identified as being redundant, complex to maintain, not integrated nationally, manually intensive, and not easily extensible (e.g. for new data types). These issues make it challenging to apply modern data management and quality assurance techniques. Furthermore, without addressing these issues, over time the system will become increasingly expensive to maintain and in some cases could compromise data integrity. In response, a new Data Management Framework (DMF) for the MSC has been conceived and embraced by the organization (Yip and Minuk 2004, 2002).

A project is currently underway to implement the DMF and achieve the following high level goals:

1. Users view the MSC data system as a single logical unit.
2. Users can obtain quality basic weather and climate data digitally on-line and in real-time.
3. Ability to monitor the operating status of all operational networks at all times.
4. Internal distributed data systems are standardized and inter-operable.

Furthermore, the following guiding principles are proposed to be performed by the core system:

1. Introduction of an "official" MSC value for any given monitoring element.
2. Once raw data are decoded, they move through the system as elements in a relational database rather than repetitive encode/decode cycles.
3. All data monitored and collected are archived.
4. Data and metadata history will be retained with no data loss.
5. Data integrity is known for all MSC monitoring sites.
6. More scientific rigor in quality assurance practices.
7. Full public access to all archived data.

To achieve the above, the DMF must incorporate a comprehensive automated quality assurance (QA) and quality control (QC) system as part of its architecture. The focus of this paper will be on this aspect of the DMF. The DMF will provide the infrastructure to support the following features aimed at better assessing and improving data quality:

1. All data to go through basic real-time QC before use by downstream components.
2. National standards and algorithms to be applied to MSC network data where appropriate.
3. Allowances for non real-time QA/QC to be performed anywhere, at any time.
4. Results from both real-time and non real-time QC are archived and accessible by all users/applications.
5. Original values have linkages to all subsequent QA/QC-related transformations and metadata.

2. QUALITY ASSURANCE AND QUALITY CONTROL

Quality is relatively easy to recognize (in products), but difficult to define. Quality means many things to different people. There are numerous interpretations and definitions for the terms quality assurance and quality control, some of which are contradictory. Upon reviewing a number of IEEE[1] documents as well as other literature (Hoyle 1994; EPA 1996; Stephens 2002; WMO,2004), the following composite definitions were formulated:

QA – *A system of planned and systematic management activities necessary to provide adequate confidence that data, products or services will fulfill quality requirements. QA includes the organization, planning, data collection, quality control, documentation, training plans, auditing, reporting, and quality improvement to ensure that quality objectives are met. QA activities establish the extent to which quality will be, is being, or has been controlled.*

QC – *A system of operational techniques and the activities that measure, assess and characterize the quality of data, products or services, through error detection and control, in order to satisfy given quality requirements. QC is a major component of total quality management and is a process for maintaining standards, not creating them*.

QA/QC – *"A system of procedures, checks, audits, and corrective actions to ensure that all technical, operational, monitoring, and reporting activities are of the highest achievable quality"* (www.epa.gov).

There exists an extensive body of literature pertaining to QA/QC procedures and systems, which will not be reviewed here. Rather, a brief overview of the features planned for implementation within the DMF will be highlighted.

An important element of a comprehensive QA system is an alerting function whose purpose is to warn users/maintainers of poor data quality, malfunctions or conditions which present a risk to health and safety (e.g. severe weather warnings). This functionality will be integrated with the various QC components of the DMF's over-all QA system. The complexity of this system does not permit elaboration here given the scope of this paper.

**Corresponding author address:* L. Dale Boudreau, National Archives and Data Management Branch, MSC, 4905 Dufferin St., Downsview, ON, CANADA M3H 5T4; email: dale.boudreau@ec.gc.ca.

[1] Institute of Electrical and Electronics Engineers.

To put the QA/QC discussion into context, there will be a brief overview of the DMF in general, followed by specific proposals for QA/QC techniques that the DMF will accommodate.

3. MSC'S DATA MANAGEMENT FRAMEWORK

The basic infrastructure of the DMF will be a component-based architecture where various modules interact with a relational database through '*middleware*' to achieve certain operations required by other modules (Figure 1). Any new transformations, data or metadata that are produced in these operations will be sent to the storage layer to maintain a complete history.

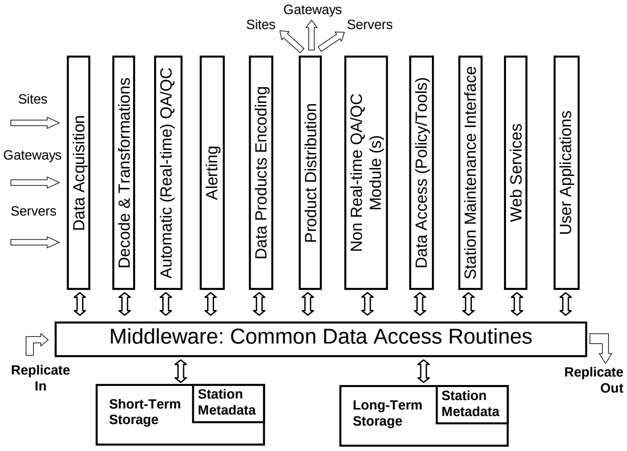

Figure 1. Component-based architecture of the DMF.

In this scheme the QA is the responsibility of several components; a real-time module to perform basic QC on point data, and several specialized modules to conduct more complex non real-time QC (e.g. network specific, spatial, trend analysis, interactive QC, etc.).

To accommodate telecommunication challenges and the sheer volume of data, several complete DMF '*nodes*' will be used as required to comprise the one logical DMF. Each node will handle data collection, decode, transformation, QA/QC, archiving, product generation, and distribution. Multiple nodes will be synchronized such that data are replicated between them and they are viewed as one logical system (Figure 2).

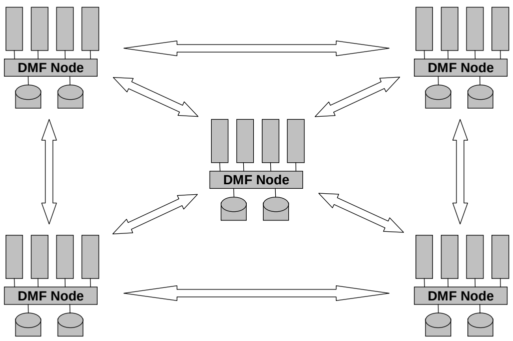

Figure 2. DMF nodes forming one logical system.

The architecture and middleware of the DMF rely on J2EE[2] for the coding and integration of modules. Given the distributed nature of the DMF nodes as well as the allowance for QA/QC procedures to be done in different locations at different times, the DMF architecture must be highly integrated; both within and between nodes.

As illustrated in Figure 1, there are various components within a DMF node that are responsible for aspects of QA which are integrated through J2EE features and the middleware. In the prototype DMF, the various modules "subscribe" to "topics" and the system ensures proper delivery of data objects and their persistence. For example, the real-time QC module would subscribe to the *DecodedData* topic and create a new output data object with a topic of *RT-QCdata*. In this example all data types which have been collected, decoded and transformed (e.g. unit conversions) are forked; with a copy stored in the database while the other is directed to the real-time QC module. Downstream the Data Products Encoder may subscribe to the *RT-QCdata* topic before creating bulletins for external distribution. In this scheme the failure of any one node/component will not adversely impact other nodes/modules, with the exception of possibly delaying data delivery.

In this distributed, but synchronized scheme, a user accessing the DMF from one node would have access to all data in the DMF. Likewise, when new data are processed by one node, it propagates throughout the entire system. This resolves the stovepipe architecture of the current system where there are many regional processes/databases that are not fully integrated into a unified national system. Such a fragmented system results in difficulties accessing data, a lack of standardization, multiple values for the same element, output from value added products and non real-time QC not being accessible nationally, and risks having multiple points of failure. The integrated approach used in the DMF resolves these issues and provides a solid foundation upon which to enhance the quality assurance of all MSC monitoring data.

The flexibility and extensibility of the system lies in the way data are delivered to the DMF components. After collection of native data formats, decoding and transformations are performed to convert data to an elemental form suitable to be databased. The entity that a data value represents is decomposed into its elemental constituents, similar to the way data are described in the WMO BUFR format (WMO 2001). For example if a particular datum represents "*average 10-minute wind speed on the hour at a height of 10 m*", the following metadata would be assigned to the data value to identify its meaning:

- Element class = Wind
- Element name = Speed
- Units = m s^{-1}
- Time period displacement = -10 minutes
- Time period duration = 10 minutes
- Statistical significance = average
- Measurement height above surface = 10 m

[2] Java 2 Enterprise Edition

The last four bullets are examples of data qualifiers which are used to decompose elements (others exist, but are not required in this example). By normalizing data across networks and message types in this way, it is possible to characterize virtually any data type in a consistent manner. This technique provides the means to subject data from different sources to the same basic QA/QC procedures.

4. QA/QC WITHIN THE DMF

The DMF must process all operational monitoring data collected by the MSC, yet it is expected that all data be subjected to real-time automatic QC. This requires that a balance be struck between complexity and simplicity since these competing requirements directly impact overall performance. The approach taken within the DMF is to perform *basic* automatic QC on all data immediately after acquisition and decode, ideally in real-time, yet maintain the flexibility in the architecture to easily accommodate near and non real-time QC modules to handle the more sophisticated quality assessment. This scheme satisfies operational requirements for near real-time data users and systems, as well as ensures the integrity of the archive for users of longer term data sets.

Of the three commonly used QA/QC methods; Sequential, Bayesian and Complex (Collins 2001), the approach used in the DMF will most closely resemble the "complex". The term complex does not necessarily mean complicated, it simply means there is a complex of several components where the component tests are independent (Gandin 1988). Decisions on data quality are not made until the results from all the tests are known. At this point a decision making algorithm assigns summary flags at various levels based on an analysis of all the test results. If there are a series of QC components (e.g. real-time, several non real-time, interactive, etc.), a decision making algorithm again assesses all the information to make revised judgments on data quality as new information becomes available.

The interoperability described in Section 3 allows the consistent application of QA/QC across networks and data types. The benefits of a unified, integrated QA/QC system include:

- avoids processing duplication;
- easier to maintain standards;
- consistent processing and output;
- easier for clients to interpret quality;
- efficient (one QC maintenance interface, QC done in one logical place with a distributed database, etc.);
- integrated QA and QC processes (e.g. Alerts); and
- one QA/QC process can handle multiple data types.

The flexibility of the system lies in the way data are delivered to the DMF components as decomposed elements. The technique of using generic metadata qualifiers to fully describe data has been mirrored by the QC test assignment scheme used by the QC modules when retrieving tests from the database for particular elements.

4.1 *The Automatic Real-Time QC Component*

The bulk of the automatic quality assessment routines are concentrated in the real-time QC module which operates on the decoded and transformed data. Since this module is required to process numerous data types, the processing logic is very generic and the application is quite basic. The module only knows how to retrieve and perform "QC Tests" for a given element. Very little business logic resides within the module and no details of QC tests are hard-coded. All the information required to perform QC is stored as test parameters in database tables.

The components of a QC Test are as follows:

1. an element under test;
2. a function/algorithm;
3. function parameters (e.g. constants, operands, associated element names, etc.);
4. any test logic/constraints regarding applicability (e.g. network, station, message type, zone, dates, etc.);
5. QC flag assignments; and
6. QC message output format.

For each data element, QC tests can be created for several categories within the real-time module. Flags are assigned to both the *true* and *false* function results. This allows the user to employ the same function for different purposes. Tests that have not "passed" will also generate a warning string which is of use during non real-time QA activities or in alerts. If a QC category has multiple tests, the flags from each test will be evaluated and rolled-up into a summary flag for that category. After all the QC tests have been run on an element, all the QC category summary flags will be rolled-up into a single over-all summary flag.

Currently there are five QC categories in the real-time module: presence, integrity, range, inter-variable and temporal. Any number of tests can be assigned to some or all QC categories. A conceptual diagram of the real-time QC module is shown below in Figure 3.

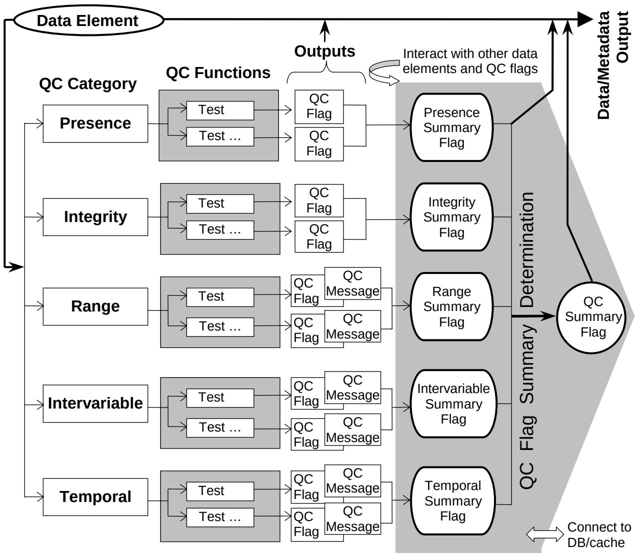

Figure 3. Conceptual diagram of the real-time QC system.

4.2 *Quality Flags*

Flags are generally used to concisely convey metadata relating to individual data values. To allow users to focus on a manageable number of flags in their area of interest, the DMF provides the ability to group flags into different categories. To satisfy different purposes or client requirements, there are categories such as Quality flags, Process flags (estimated, corrected, derived, original value, unit conversion, etc.), Warning/Diagnostic flags (value suppressed, test data, hardware state, etc.) and so on. Adding new categories or individual flags is simply a matter of database table entries.

The Quality category flags are used as qualitative indicators representing the level of confidence in the data. Currently a very basic and simple flagging scheme is used in the real-time QC module. Having too many gradations of data quality flags risks making the quality assessment vague or meaningless to users. The quality flags currently used are given in Table 1.

Table 1. Quality flags used in the real-time QC module.

Flag Value	Description	Abbreviation
-1	Missing	M
0	Error	E
10	Doubtful	D
20	Inconsistency	I
100	Accepted/Passed	A

The missing flag indicates that a particular element that was expected to be in the observation (i.e. mandatory) was either not present, had no value, or had a code indicating missing data. An error flag gets assigned to values that are either physically impossible, outside the measurement capability of the sensor, or far outside the historically observed extremes. A doubtful flag is assigned to values outside the nominal accuracy/operating limits of the sensor or approaching historical extremes. Although the observed values may still be theoretically possible, the accuracy may be compromised and users should exercise caution or re-evaluate with additional quality information. The inconsistency flag warns that a relationship between different elements does not satisfy defined criteria. Inter-variable checks do not normally have the ability to identify which element is incorrect; the inconsistency flag merely identifies a discrepancy between associated elements. Finally, accepted/passed simply indicates that a value has passed a QC test. The terms *acceptable* or *passed* are used in place of “good” or “okay” because an element passing a test does not necessarily mean it is correct.

4.3 *Functionality within the QC Categories*

Tests in the presence category ensure that the value field has contents (e.g. not blank, null, empty string, etc.) for a given element. In some cases tests may analyze the *value* to check if it is actually a code which represents “missing”. Failed tests in this category would receive a Missing flag.

Tests in the integrity category evaluate the data *format* of elements to determine whether it complies with expected data types (e.g. string, alphanumeric, integer, real, etc.). For some elements, tests may also verify that the value/string is a member of a specified set. Other tests may check the unit field of the element to ensure the incoming unit matches that which tests in other QC categories expect. Failed tests in this category would receive an Error flag.

Tests in the range category determine whether observational values lie within specified ranges (e.g. $x < 100$; $-50 \leq x \leq 50$, etc.). In addition to basic range tests which are applied to various levels of specificity, a more elaborate scheme termed an “optimized range test” is available for use within the range QC category. Tests of this nature use range parameters that have been optimized for several value ranges of an associated parameter (which has itself previously undergone basic QC and was deemed acceptable). For example, to select parameters for an optimized range test on pavement temperature, the system would: i) obtain the value of a validated air temperature within the same observation, ii) determine what range it was in, and iii) select the pavement temperature range parameters that are appropriate for that particular range of air temperatures. The approach taken of using optimized range parameters is very powerful but requires a lot of historical data analysis to characterize the relationships. Currently, failed tests in the range category receive an Error or Doubtful flag.

Tests in the inter-variable category directly compare the value of an element under test to that of an associated element(s). For example, a test may ensure that the dew point temperature is less than or equal to the ambient temperature. Similar to the optimized range test, the associated element must first have gone through basic QC and been deemed acceptable before proceeding with the comparison. Failed tests in this category receive an Inconsistency flag because the comparison was not consistent with the expected outcome and it can be difficult to determine which element is responsible for the problem.

Tests in the temporal category are of the basic variety for the real-time module. The aim of these tests is to analyze the rate of change of an element’s value between successive observations to identify “flat-lining” (persistence), unrealistic spikes, and uncharacteristic sustained drops or jumps (steps). Persistence tests will look at the absolute value of the difference between successive values and compare this to an allowable tolerance (near zero tolerances are used for identifying flat-lines). Another parameter of temporal tests is the duration for which values are allowed not to vary. Other tests to detect spikes will ensure that the absolute value of the difference between successive values for a given interval is below a certain threshold. Failed tests in the temporal category can receive an Error or Doubtful flag, depending on the situation.

More sophisticated temporal tests (e.g. step tests, trend analysis, pattern matching, etc.) as well as spatial tests will be handled by non real-time modules given their complexity, requirement for longer data sets and increased database usage.

4.4 *QC Test Assignment flexibility*

Some of the metadata stored in the database as part of a QC test also serves to determine the level of specificity. This allows the basic tests described above to be applied to a wide range of levels as well as multiple elements. In well defined networks with standardized installations and instruments, many of the QC tests will be assigned at the network-element level, avoiding defining QC tests specifically for all stations. Furthermore, since tests are broken down in a similar element decomposition scheme as that used during DMF decoding (see Section 3), where appropriate, tests may target elements in a general sense and propagate to all their derivatives. For example, a coarse test may be designed to ensure wind speed for a particular network is ≥ 0 and < 60 m s^{-1}. This one test could apply to all stations in the network and propagate to any elements that satisfy the criteria of being a wind speed (e.g. average hourly wind speed at 10 m, maximum hourly wind speed at 10 m, instantaneous wind speed at 2 m, 10-minute average wind speed at 10 m, etc.).

To better take into account the effects of large-scale and local climatology, several options exist. First of all, network-element level tests can be refined by incorporating seasonality into the tests based on their effective dates. By having tests with different start/end effective dates, tests can be created for specific months, seasons, etc., each with parameters optimized for those conditions. To address local climatology, tests can be assigned to the network-**station**-element level to use test parameters that are tuned for the unique conditions that exist at that site. To avoid having to define many specialized tests for individual stations, a "zone" parameter is available as a test option to take advantage of cases where stations can be grouped into specific climatic zones, geographical regions, drainage basins, forecast regions, etc.

Using the test assignment scheme of specifying various optional pieces of metadata as test parameters, it is possible to target QC tests all the way down to the instrument level. This gives quality assurance administrators great flexibility but also presents challenges for the population and maintenance of the QC test data. As the project evolves, it will become necessary to have interactive maintenance screens to input, visualize and manipulate the QC test information.

4.5 *Performing Real-Time QC Tests*

When a data object containing an observation (collection of data elements) arrives at the QC module, the system checks the database (or cache) for applicable tests for each element and returns them to the module. For each element under test, all basic QC tests are run in the following sequence:

1. **Presence** tests;
2. if value present, **Integrity** tests;
3. if value present and format okay, basic **Range** tests;
4. if value present and format okay, **Temporal** tests;
5. above repeated for all elements in the observation.

After all QC tests are completed in a particular QC category, there is a summary flag determination; a) for the category (simply the most severe test flag assigned), and b) the over-all summary flag for the element (the most severe of all the category summary flags). In Table 1, decreasing flag values indicate increasing levels of severity.

After the first pass through the observation, a second pass is made for any element tests which rely on a valid associated element:

6. for elements whose summary flag is not -1 or 0, perform any **optimized Range** tests they may have;
7. update summary flags as needed;
8. for elements whose summary flag is not -1 or 0, perform any **Inter-variable** tests they may have;
9. update summary flags (categorical and over-all) as needed.

The data object that leaves the real-time QC module contains all the original observation's data and metadata with the addition of all the QC results for each element, namely individual test results (flags and any error messages), category summary flags, and the over-all summary flag. As the DMF develops, the simplistic algorithm for summarizing flags may evolve and become more sophisticated, although for the real-time QC module the speed and simplicity of the algorithm is appropriate. Subsequent to real-time QC, if any non real-time modules operate on the observation (e.g. trend checks, spatial checks, etc.), there is another more sophisticated summary flag determination. In this way, the quality assessment of elements is continually updated throughout the QA process as more information becomes available.

4.6 *Non Real-Time QC Components*

Although the real-time QC module is the focus of development in the early stages of the DMF's implementation, the architecture allows for other QA components to be added as needed. Spatial tests, where elements are compared to similar data at neighboring sites for the same time period, cannot be performed in real-time due to the possibility of varying raw data receipt times or processing delays. Also complex inter-variable tests, which compare an element to proxy data from other data sources/networks (e.g. comparing rainfall data to quantitative precipitation estimates derived from weather radar data), will have to be scheduled to accommodate different reporting intervals. For these QC procedures, as well as more complex temporal tests which perform trend checks over lengthy time periods, non real-time QC modules will be required. For certain data types which receive funding to perform manual QC, graphical user interface modules will be developed to facilitate interactive data correction.

The above components, and others used to handle QA activities such as alerting, will be developed over the next several years as the DMF project matures. Non real-time modules which perform QC functions will be integrated with their real-time counterparts. As each process supplies additional quality assessments and flags for data elements, the over-all summary flag will be re-evaluated and possibly changed. When new test results and summary flags are generated for data elements, the metadata will be re-circulated throughout the DMF so that data users and product generators will benefit from the latest quality assessments.

5. SUMMARY

The MSC is in the early stages of implementing a new approach to data management which will address issues and deficiencies encountered in the present system. The Data Management Framework project has been undertaken to implement a system to better manage environmental monitoring data. The principles of the DMF and its component based architecture allow for enhancements to the quality assurance of data which were previously not possible or easily achieved.

The DMF, with its high degree of integration between distributed nodes as well as between real-time and non real-time QC modules, will allow there to be a single authoritative data value and assessment of quality for each element, for any given point in time. This feature allows all clients accessing the DMF to obtain consistent data/quality results. As various quality assessments are made through time by real-time, non real-time and interactive QC modules, decision making algorithms evaluate all the quality information to render a judgment on the over-all validity of data elements.

The increased efficiency and automation of the QA system within the DMF will result in improved data quality assessments being made in a more timely manner compared to what the current system is capable of now. This benefit, coupled with an integrated alerting system, will form an important link in a positive feedback loop to improve over-all MSC data quality at source.

6. ACKNOWLEDGEMENTS

The authors would like acknowledge the work of developers in Québec Region who built a regional data management system (BDQ[3]) that incorporates many of the concepts that the DMF is striving to implement. Figure 3 in this paper is a modified version of a conceptual diagram describing the QA/QC process within BDQ.

7. REFERENCES

Collins, W.G., 2001: The operational complex quality control of radiosonde heights and temperatures at the National Centers for Environmental Prediction. Part I: Description of the method. *J. Appl. Meteor.*, **40**, 137 - 151.

EPA, 1996: *The Volunteer Monitor's Guide to Quality Assurance Project Plans: Chapter 3 – Some Basic QA/QC concepts*, EPA 841-B-96-003, pp. 59.

Gandin, L.S., 1988: Complex quality control of meteorological data. *Mon. Wea. Rev*., **116**, 1137 - 1156.

Hoyle, D., 1994: *ISO9000 Quality Systems Handbook*. 2nd Ed., Pub. Butterworth-Heinemann, London, pp. 420.

Stephens, K., 2002: *The Best on Quality – Vol. 13.* ASQ Quality Press, Milwaukee, Wisconsin, pp. 397.

WMO, 2004: *Commission For Basic Systems– Open Programme Area Group On Integrated Observing Systems – Expert Team On Requirements For Data From Automatic Weather Stations*. Geneva, June 28 – July 2, 2004. ET-AW-3, Final Report, Annex 5, p. 2.

WMO, 2001: *Manual on Codes – International Codes: Vol. I.2 Part B.* WMO Publication No. 306, Geneva, Switzerland.

Yip, T.C. and M. Minuk, 2004: Data Management Framework of the Meteorological Service of Canada, *Proceedings of the 20th International Conference on Interactive Information Processing Systems.* 11-15 January 2004, Seattle, Washington. (CD-ROM).

Yip, T.C. and M. Minuk 2002: Future of the data holdings of the Meteorological Service of Canada, *Proceedings of the 18th International Conference on Interactive Information Processing Systems*. 13-17 January, 2002, Orlando, Florida, 277-278.

8. Links

ASQ (American Society of Quality):
http://www.asq.org/topics/qa_qc.html

EPA (Environmental Protection Agency):
http://www.epa.gov/OCEPAterms/qterms.html

[3] BDQ - Banque de Données Qualifiées (Qualified Data Base)