

## Quality Control of Environmental Measurement Data with Quality Flagging

Mauno Rönkkö, Okko Kauhanen, Markus Stocker, Harri Hytönen, Ville Kotovirta, Esko Juuso, Mikko Kolehmainen

► **To cite this version:**

Mauno Rönkkö, Okko Kauhanen, Markus Stocker, Harri Hytönen, Ville Kotovirta, et al.. Quality Control of Environmental Measurement Data with Quality Flagging. Ralf Denzer; Robert M. Argent; Gerald Schimak; Jiří Hřebíček. 11th International Symposium on Environmental Software Systems (ISESS), Mar 2015, Melbourne, Australia. Springer, IFIP Advances in Information and Communication Technology, AICT-448, pp.343-350, 2015, Environmental Software Systems. Infrastructures, Services and Applications. <10.1007/978-3-319-15994-2\_34>. <hal-01328569>

**HAL Id: hal-01328569**

**<https://hal.inria.fr/hal-01328569>**

Submitted on 8 Jun 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



# Quality Control of Environmental Measurement Data with Quality Flagging

Mauno Rönkkö<sup>1</sup>, Okko Kauhanen<sup>1</sup>, Markus Stocker<sup>1</sup>, Harri Hytönen<sup>2</sup>,  
Ville Kotovirta<sup>3</sup>, Esko Juuso<sup>4</sup>, and Mikko Kolehmainen<sup>1</sup>

<sup>1</sup> Department of Environmental Science,  
University of Eastern Finland, Kuopio, Finland  
{mauno.ronkko, okko.kauhanen, markus.stocker, mikko.kolehmainen}  
@uef.fi

<sup>2</sup> Vaisala Oyj, Vantaa, Finland  
harri.hytonen@vaisala.com

<sup>3</sup> VTT Technical research Centre of Finland, Espoo, Finland  
ville.kotovirta@vtt.fi

<sup>4</sup> Faculty of Technology, Control Engineering,  
University of Oulu, Oulu, Finland  
esko.juuso@oulu.fi

**Abstract.** We discuss quality control of environmental measurement data. Typically, environmental data is used to compute some specific indicators based on models, historical data, and the most recent measurement data. For such a computation to produce reliable results, the data must be of sufficient quality. The reality is, however, that environmental measurement data has a huge variation in quality. Therefore, we study the use of quality flagging as a means to perform both real-time and off-line quality control of environmental measurement data. We propose the adoption of the quality flagging scheme introduced by the Nordic meteorological institutes. As the main contribution, we present both a uniform interpretation for the quality flag values and a scalable Enterprise Service Bus based architecture for implementing the quality flagging. We exemplify the use of the quality flagging and the architecture with a case study for monitoring of built environment.

**Keywords:** quality control · quality flagging · enterprise service bus · environmental data · built environment

## 1 Introduction

Environmental measurement and monitoring has been a growing trend for the past decade [1]. It is needed for instance for assessing the negative impact of human activities to the environment [2,3]. Environmental measurements, however, are prone to external variation and even disruptions. Therefore, raw measurement data must always be somehow preprocessed before it can be used in computations as an input. There exist standards for the representation of environmental data. For instance, the

Open Geospatial Consortium provides standards for the representation and access of the spatial data. However, the standards do not address the issue of data quality comprehensively. For instance, UncertML [4] was proposed as an extension to OGC, to address uncertainty representation. With UncertML, one can attach probabilistic uncertainties to environmental data sets. Still, standards do not provide sufficient support, for instance for real-time quality control of environmental data, as discussed in Section 2.

Quality flagging is a means to provide quality information on the level of individual measurement data points both in real-time and off-line. Most importantly, quality flagging is also a reversible activity, as it preserves all original measurement values. As discussed in Section 2, we focus on one specific quality flagging scheme. It is the scheme presented by Vejen et al. [5] that is recommended and used by the Nordic meteorological institutes. Since the quality flagging scheme by Vejen et al. is tailored for weather measurement data, we propose as part of the main contribution a uniform interpretation for the quality flag values to be used for the flagging of any kind of environmental measurement data. It should be noted that the quality flagging scheme by Vejen et al. is not what World Meteorological Organization (WMO) refers to when speaking of a Quality Management Framework. In particular, WMO strives after an ISO certification, whereas the quality flagging scheme is a technical implementation of a real-time and off-line computational quality procedure.

As the other part of the main contribution, in Section 3, we present an Enterprise Service Bus (ESB) [6] based architecture to perform quality flagging in a scalable and measurement device independent manner. In Section 4, we illustrate the use of the ESB based architecture to perform quality flagging of data for built environment, including room temperature and water consumption measurements. As the research work is still ongoing, we present here our complete plans. We conclude in Section 5.

## 2 Quality Control of Environmental Measurement Data

When considering quality control of environmental measurement data, there is practically one well-known proposed standard, UncertML [4]. With UncertML, one can attach probabilistic uncertainties to environmental data sets to support statistical pre-processing. For instance, Williams et al. [7] used UncertML to attach uncertainty information to raw weather data as provided by Weather Underground<sup>1</sup>. By using UncertML and INTAMAP [8], they were able to estimate the bias and residual variance, to adjust, merge, and interpolate temperature data from independent data sources. As a result, they were able to produce an interpolated temperature map for the whole UK based on the Weather Underground data with statistical corrections.

Although UncertML does provide means to improve the quality of environmental data, it operates on the level of measurement data sets. Such level of data quality, however, is not sufficient for all applications. A complementary approach is to per-

---

<sup>1</sup> <http://www.wunderground.com>

form quality control on the level of individual measurement data points. For this purpose quality flagging is used.

**Table 1.** Quality flag values and their original interpretation [5] along with the proposed generic interpretation.

Flag	Original interpretation	Generic interpretation
0	No check performed	Value not checked
1	Observation is ok	Approved value
2	Suspected small difference	Suspicious value
3	Suspected big difference	Anomalous value
4	Calculated value	Corrected value
5	Interpolated value	Imputed value
6	(Not defined originally)	Erroneous value
7	(Not defined originally)	Frozen value
8	Missing value	Missing value
9	Deleted value	Deleted value

The Nordic meteorological institutes have developed a fully functioning quality flagging scheme as discussed by Vejen et al. [5]. It provides both real-time and off-line quality flagging. Vejen et al. [5] distinguish between four quality control levels. QC0 is a real-time quality control performed by the measurement devices or stations. QC1 is a real-time quality control performed by the data acquisition system prior to storing the data. QC2 is an off-line quality control performed by the data management system based on the stored data. Lastly, HQC is the final off-line quality control check performed by a human operator. Each of these levels use the same quality flag values as indicated in **Table 1**. Thus, the quality flag is a number with four digits:  $C = E_{QC0} + 10 \times E_{QC1} + 100 \times E_{QC2} + 1000 \times E_{HQC}$ , where each  $E_{QC0}$ ,  $E_{QC1}$ ,  $E_{QC2}$  and  $E_{HQC}$  are quality flag values for the corresponding quality control levels.

Because the quality flagging scheme by Vejen et al. is designed for weather measurements, it does not apply to generic environmental measurement data. In particular, the original interpretation can be non-informative or misleading in a generic case. Also, the original flag values do not support observations of a malfunctioning measurement device that produces constant, “frozen”, or clearly erroneous measurement values. Therefore, we propose a generic interpretation for the quality flag values, as indicated in **Table 1**. The proposed interpretation is downwards compatible with the original interpretation, so that it could also be used for weather measurements. In particular, in the generic interpretation “suspicious value” and “anomalous value” are used instead of “small difference” and “big difference”. Also, “imputation” is used instead of “interpolation”, as interpolation may not be applicable in a generic case. Similarly, the generic interpretation replaces “calculated value” with “corrected value” to emphasize the difference between value correction and missing value imputation. Lastly, the generic interpretation uses the two originally unused flag values for diagnostics, to indicate a clear measurement error or a “frozen” measurement value.

It should be noted that quality flagging is a complementary approach with respect to use of UncertML. In particular, quality flagging operates on the level of individual measurement points, whereas UncertML operates on the level of data sets. Thus, both can be applied to the same data set at the same time to provide detailed information about the data quality. What makes the quality flagging an attractive approach is that it supports quality restrictions during the data queries. For instance, one can query only such data, where the final quality check has been performed. Similarly, one can query data, where there are no QC0 or QC1 failures or corrections. Implementing such queries requires no extra work, as they can be constructed based on the quality flag values. Such a query style is supported, for instance, by all SQL databases. Furthermore, queries about failures, suspicions, and corrections provide valuable information to be used with UncertML. In particular, information about bad quality can be used to select and fine tune appropriate statistical and probabilistic model for UncertML, to match the observed data quality.

Quality flagging provides also valuable information to systems diagnostics and maintenance. The frequency and trend of quality failures function as indicators for device failures or model inadequacies. Thus, the bigger the measurement network is, the more useful and valuable quality flagging becomes. This is something that is not currently addressed by methods such as UncertML that focus on interoperability at the level of datasets.

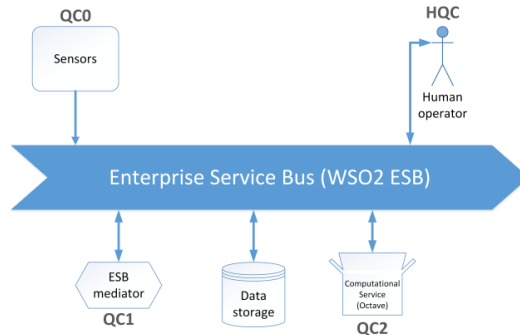
### 3 The Enterprise Service Bus Based Architecture

As depicted in **Fig. 1**, the role of the ESB is to pass measurement data as messages between services. More specifically, we use here WSO2 ESB<sup>2</sup>. The ESB is extended and configured so that it has a dedicated port and a mediator for each sensor. Thus, a sensor performs QC0 on the measurement data after each measurement and sends the data in its native format, such as JSON or XML, to the dedicated ESB port. The ESB then redirects the received measurement data to a dedicated mediator that performs QC1 and passes the checked measurement data back to the ESB. The ESB then redirects the checked measurement data to a data storage. The ESB is configured to trigger QC2 on stored data on regular intervals. The actual QC2 is then performed by a computational service, for which we use Octave<sup>3</sup>. For this purpose, a predetermined subset of the stored data is retrieved for the computational service. After QC2, the checked data is stored back to the data storage. Lastly, HQC is performed by a human operator on the data that is already checked by QC2. HQC is initiated by the human operator through a dedicated client application. The client application accesses the stored data requested by the operator through the ESB. Similarly, after HQC, the client application stores the checked data back to the data storage through the ESB.

---

<sup>2</sup> <http://wso2.com/products/enterprise-service-bus/>

<sup>3</sup> <https://www.gnu.org/software/octave/>



**Fig. 1.** An Enterprise Service Bus based architecture for quality flagging.

The advantage of the ESB architecture is that it can be reconfigured by an administrator while the system is running. Therefore, it is possible to add new sensors and algorithms, as well as expand and refactor a data storage while the system is in use. The reception ports of the ESB can also be configured to receive measurement data as messages virtually in any format. For instance, the WSO2 ESB supports by default messages that are passed in HTTP and SOAP format. It should be noted that the ESB architecture is scalable by using multiple ESB instances to improve performance by passing the messages in between them. Several concurrent instances of the ESB architecture can also be used to make the overall system more robust and fault-tolerant.

The ESB architecture also supports the use of OGC SWE Standards. The ESB architecture can be extended with ports configured, for instance, to receive and pass OGC O&M compliant data. Similarly, the architecture can be extended to support the OGC SOS standard for sensor data management.

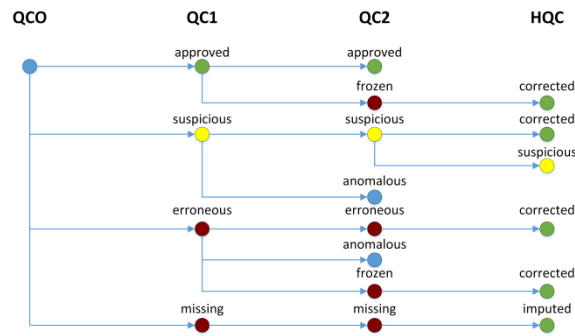
## 4 Case Study

As our case study, we consider the monitoring of residential buildings. The buildings sector is the largest user of energy and CO<sub>2</sub> emitter in the EU, estimated at approximately 40% of the total consumption. In particular, we study a specific home monitoring system called AsTEKa [9,10]. For simplicity, we consider here only two variables: room temperature and water consumption. The sensors for these two variables are both physically and technologically different. As the research work is still ongoing, we present here our complete plans. We have already implemented the ESB architecture and we have studied various statistical methods to be used in the quality flagging. We have also implemented example mediators for quality control, but we have not yet implemented in full the quality flagging scheme that we discuss next.

We decided not to consider all quality flag values for all quality control checks. Instead, only the most critical quality checks are considered, as indicated in **Table 2**. In particular, QC0 was not used, as AsTEKa uses low-cost sensors that do not support real-time computations. Instead, QC1 is extended to consider also the checks usually performed by QC0.

**Table 2.** Quality flag values used in AsTEKa quality control; an applicable flag value is indicated by “yes”.

Flag	Interpretation	QC1	QC2	QC3	HQC
0	Value not checked	no	no	yes	yes
1	Approved value	no	yes	yes	yes
2	Suspicious value	no	yes	yes	yes
3	Anomalous value	no	no	yes	yes
4	Corrected value	no	no	yes	yes
5	Imputed value	no	no	yes	yes
6	Erroneous value	no	yes	yes	no
7	Frozen value	no	no	yes	no
8	Missing value	no	yes	yes	no
9	Deleted value	no	no	yes	yes



**Fig. 2.** Quality control of room temperature data.

**Room temperature.** Fig. 2 depicts the whole chain of quality control for room temperature data. QC1 decides coarsely if the data points are approved, suspicious, erroneous, or missing. As QC1 runs once a minute, quality controlled data points are available one minute after the measurement. Hence, such approved data points can be used for near real-time control of heating and cooling. By avoiding using suspicious and erroneous data points we can also avoid unnecessary heating and cooling.

Since QC2 runs every 2 hours, data after QC2 can be used for alerting occupants and maintenance personnel of anomalies and potential malfunctions. When considering heating and cooling, a 2 hours window is sufficient to prevent systemic failures that could cause damage to devices or structures. Thus, data after QC2 is particularly suited for diagnostic purposes and detecting occupant behavior or system settings causing to waste energy.

As HQC aims at resolving frozen, erroneous, or missing data, it is useful for analyzing structural changes in the residential building. Because structures weaken over time and the performance of heating or cooling devices also deteriorates over time, one can expect an increasing trend in use of energy over time. This trend can be computed by comparing quality controlled room temperature values with use of heating

and cooling energy. As such a change is not abrupt, it is sufficient to perform HQC once a month. The frequency and number of performed corrections by HQC acts also as an indicator for the condition of the home monitoring system as a whole.

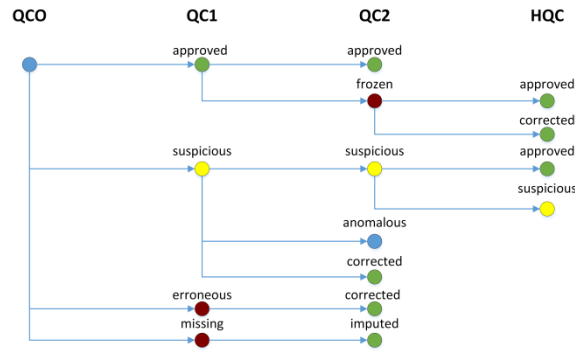


Fig. 3. Quality control of water consumption data.

**Water consumption.** Fig. 3 depicts the whole chain of quality control for water consumption data. QC1 decides coarsely if the data points are approved, suspicious, erroneous, or missing. As QC1 runs once a minute, only the erroneous and missing data points are of interest. In such a case, maintenance personnel could be notified and the measurement devices could be repaired quickly.

Since QC2 runs every 2 hours, data after QC2 can be used for alerting occupants and maintenance personnel of anomalies and potential malfunctions. In particular, the data after QC2 can be used to spot leaks and malfunction of valves and appliances that use water. When considering water consumption, a 2 hours window is generally sufficient to prevent systemic failures that could cause damage to appliances or structures. Thus, data after QC2 is particularly suited for diagnostic purposes and detecting occupant behavior or system malfunctions causing to waste water or causing structural damage. The frequency and number of performed corrections by QC2 indicates also the condition of the home monitoring system as a whole.

As HQC aims at resolving remaining frozen data, it is useful for analyzing the condition of appliances as well as occupant behavior that leads to wasting water. As such conditions do not evolve fast over time, it is sufficient to perform HQC once a month.

## 5 Conclusion

We studied the use of quality flagging as a means to perform quality control of environmental measurement data. We proposed the adoption of the quality flagging scheme introduced by the Nordic meteorological institutes to be used with any kind of environmental measurement data. We presented both a uniform, generalized interpretation for the quality flag values and a scalable Enterprise Service Bus based architecture for implementing the quality flagging. We exemplified the use of the quality



flagging a case study for the monitoring of built environment. Our research is ongoing. We presented our design and approach to quality control by quality flagging. We have implemented the core ESB based architecture and we are currently implementing the quality flagging algorithms.

As for future work, we plan on studying occupant profile-based imputation of missing or erroneous values. Such profiling based models could also be used earlier in quality control, for instance, by having QC1 flagging suspicious values with respect to profile based reference values. We also plan on including different kinds of measurement variables, such as CO<sub>2</sub> and humidity. This would enable monitoring indoor air quality and automated notifications on degraded air quality. We are also simultaneously investigating the use of quality flagging in a sensor network monitoring water quality of lakes in Finland together with the Finnish Environment Institute.

## Acknowledgements

We wish to thank the anonymous reviewers for valuable comments. This research is funded by the Tekes project “MMEA: Measurement, Monitoring, and Environmental Assessment” (Decision number 427/10).

## References

1. H. Messer, A. Zinevich, and P. Alpert. Environmental Monitoring by Wireless Communication Networks, *Science*, vol. 312, pp. 713, 2006.
2. D. Cyranoski, and G. Brumfiel. Fukushima impact is still hazy, *Nature*, vol. 477, pp. 139-140, 2011.
3. N. Gilbert. Drug waste harms fish, *Nature*, vol. 476, pp. 265, 2011.
4. M. Williams, D. Cornford, L. Bastin, and E. Pebesma. *Uncertainty markup language (UncertML)*, OpenGIS Discussion Paper 08-122r2, Open Geospatial Consortium Inc., 2009.
5. F. Vejen (ed), C. Jacobsson, U. Fredriksson, M. Moe, L. Andresen, E. Hellsten, P. Rissanen, T. Palsdottir, and T. Arason. *Quality Control of Meteorological Observations. Automatic Methods Used in the Nordic Countries*. Climate Report 8/2002, Norwegian Meteorological Institute, 2002.
6. D. A Chappell. *Enterprise Service Bus: Theory in Practice*, O'Reilly, 2004.
7. M. Williams, D. Cornford, L. Bastin, R. Jones, and S. Parker. Automatic processing, quality assurance and serving of real-time weather data, *Computers & Geosciences*, 37, pp. 351-362, 2011.
8. M. Williams, D. Cornford, B. Ingram, L. Bastin, T. Beaumont, E. Pebesma, G. Dubois. Supporting interoperable interpolation: the INTAMAP approach. *In Proceedings of the International Symposium on Environmental Software Systems*, Prague, 2007.
9. J-P. Skön, O. Kauhanen and M. Kolehmainen, Energy Consumption and Air Quality Monitoring System, *Proc. of the 7th International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pp. 163-167, 2011.
10. J-P. Skön, M. Johansson, O. Kauhanen, M. Raatikainen, K. Leiviskä, and M. Kolehmainen. Wireless Building Monitoring and Control System, *World Academy of Science, Engineering and technology*, 65, pp. 706-711, 2012