Wolfgang Lehner, Anja Klein

Representing Data Quality in Sensor Data Streaming Environments

Erstveröffentlichung in / First published in:

Journal of Data and Information Quality. 2009, 1(2), S. 1-28 [Zugriff am: 20.05.2022]. ACM. ISSN 1936-1963.

DOI: https://doi.org/10.1145/1577840.1577845

Final edited form was published in "Journal of Data and Information Quality". 2009, 1(2), S. 1-28. ISSN 1936-1963.
https://doi.org/10.1145/1577840.1577845

Representing Data Quality in Sensor Data Streaming Environments

A. KLEIN,
SAP Research CEC Dresden, Germany

W. LEHNER
University of Technology Dresden, Germany

Sensors in smart-item environments capture data about product conditions and usage to support business decisions as well as production automation processes. A challenging issue in this application area is the restricted quality of sensor data due to limited sensor precision and sensor failures. Moreover, data stream processing to meet resource constraints in streaming environments introduces additional noise and decreases the data quality. In order to avoid wrong business decisions due to dirty data, quality characteristics have to be captured, processed, and provided to the respective business task. However, the issue of how to efficiently provide applications with information about data quality is still an open research problem.

In this article, we address this problem by presenting a flexible model for the propagation and processing of data quality. The comprehensive analysis of common data stream processing operators and their impact on data quality allows a fruitful data evaluation and diminishes incorrect business decisions. Further, we propose the data quality model control to adapt the data quality granularity to the data stream interestingness.

Additional Key Words and Phrases: Data stream processing, data quality, smart items

Final edited form was published in "Journal of Data and Information Quality". 2009, 1(2), S. 1-28. ISSN 1936-1963.
https://doi.org/10.1145/1577840.1577845

1. INTRODUCTION

In smart-item environments, data concerning product usage and environmental data is captured using a multitude of sensors (e.g., pressure, temperature, mileage). The recorded sensor data streams are exploited to support and optimize production automation processes as well as complex business decisions. For example, oil condition monitoring is crucial to ensure the proper operation of hydraulic systems. A wide range of sensors, for example, pressure, oil viscosity, or particle contamination, are used to control the aging of hydraulic systems to predict efficient maintenance dates for oil or spare part exchanges.

Sensors deliver numerical, discretized, and digitized data streams. The underlying measurement process as well as sensor failures or malfunctions may lead to falsified, wrong, or missing values.

To extract complex knowledge, sensor data is merged, transformed, and aggregated by applying traditional data stream queries (defined via the Continuous Query Language, CQL), complex signal analysis, or elementary numerical operators. Data aggregation and sampling (e.g., during load shedding) are used to reduce the data volume to meet memory and communication capacity constraints in streaming environments.

During the data stream processing, the initial sensor-inherent errors are amplified. Additionally, new errors may be introduced. Finally, if the sensor data are incorrect or misleading, derived decisions are likely flawed.

There are two approaches to handle data quality deficiencies. The optimistic approach relies on sensors with high precision and assumes that the arising errors are small enough to be negligible in the application context. However, this approach requires very high costs for sensors, sensor shielding, and reliable data transfer with high communication effort. The use of redundant sensors constitutes a special case of the optimistic approach, where the very high costs for multiple sensors are justified to cover the breakdown of one sensor or to improve the initial quality by averaging multiple sensor streams. However, even so, not all sensor-inherent errors will be avoided. To prevent obsolete high costs while still guaranteeing prudent data quality management, we pursue another strategy. We carefully survey data quality restrictions in sensor data streams to allow a comprehensive data evaluation. Therefore, data quality information has to be recorded at the sensor nodes, propagated through the data processing, and finally presented to the user. For example, this allows to prevent the corrosion of hydraulic components like filters or sealings due to undetected contaminated oil, which would otherwise lead to high maintenance costs or even to a system breakdown.

Since sensors allow for the automatic collection of a huge volume of data, the additional propagation of data quality information results in an overhead for data transfer and management, which may shape up as very expensive. Furthermore, lest data quality information is lost, the executed data processing steps have to be mirrored in a data quality processing framework. Our contributions in the context of data quality management and processing are as follows.

Final edited form was published in "Journal of Data and Information Quality". 2009, 1(2), S. 1-28. ISSN 1936-1963.
https://doi.org/10.1145/1577840.1577845

(1) We present a comprehensive analysis of the impact of stream processing operators derived from CQL querying as well as signal analysis on data quality information. We propose an operator classification according to the conducted data manipulation to group data quality processing theorems.
(2) We present the data quality model control to automatically adapt the granularity of data quality information to the current data stream interestingness.
(3) We evaluate the proposed data quality operators by comparing the true error with the estimated data quality. We show the benefits of the data quality model control regarding communication capacity constraints in a data streaming environment.

This article is organized as follows: While Section 2 summarizes related work, we describe the data quality management and discuss relevant DQ dimensions in Section 3. Section 4 classifies and defines the data quality processing for a distinct set of common data stream operators derived from CQL and signal processing. Thereafter, Section 5 presents the data quality model control to find a trade-off between efficient DQ transfer and data interestingness. Section 6 provides the evaluation of the proposed data quality processing theorems and automatic model control. This work concludes with a short summary.

2. RELATED WORK

Data Quality (DQ) management in databases and data warehouses is motivated and discussed in several publications [Strong et al. 1997; Orr 1998; Wand and Wang 1996; Mielke et al. 2005]. The focus lies on definitions of the term data quality and different sets of relevant data quality dimensions. In general terms, data quality describes the suitability of data for the respective data processing application. To evaluate this suitability, Wang and Strong [1996] empirically analyzed different semantic categories of data quality, such as intrinsic, representational, and contextual quality as well as accessibility. Weikum [1999] used an application- or process-oriented approach to classify data quality dimensions. In the context of information integration on the Internet, Naumann and Rolker [1999] defined a comprehensive set of data quality definitions based on a requirement survey. Adapted from these data quality classifications, data quality definitions can be derived for various application scenarios according to the particular application requirements and priorities. The data quality of data streaming environments given in Section 3 is defined based on the intrinsic and contextual data quality presented by Wang and Strong [1996].

Despite the fact that data quality is identified as important in the context of relational databases and data warehouse environments [Burdick et al. 2005; Ballou and Tayi 1999; Motro and Rakov 1997], prior work suffers from the major drawbacks that either an active participation of users or domain

Final edited form was published in "Journal of Data and Information Quality". 2009, 1(2), S. 1-28. ISSN 1936-1963.
https://doi.org/10.1145/1577840.1577845

experts in the data quality assessment is necessary [Naumann and Rolker 2000] or that the presented approaches refer to a (set of) reference data source(s) containing the true data to calculate the data quality. The same holds for data cleaning strategies to improve data quality.

It is obvious that in case of sensor data, subsequent manual data quality estimation or correction of each measurement item is not feasible; furthermore, a high-quality reference for comparison is not present. Instead, the data quality has to be recorded at the sensor nodes and propagated through the data processing to the data consuming end application(s). Klein et al. [2007] propose a data stream metamodel extension as a basis for the data quality management. To reduce the data overhead produced by the data quality transfer, jumping data quality windows are introduced to propagate quality information not for every single measurement value but rather aggregated over a certain period of time. Additionally, Klein et al. present methods for the data quality recording as well as a metamodel extension to store data quality information in a relational database.

Aside from the management of data quality information as streaming metadata, a data quality algebra is required to track the influence of the data processing on data quality. There exist several approaches to define a data quality algebra for relational database operators, with the focus on different data quality dimensions as well as operator classes.

Motro and Rakov [1996] concentrate on soundness and completeness on relation or database level derived from the information retrieval measures *precision* and *recall*. While the completeness applies to streaming data, the soundness is not applicable to sensor data quality because every measurement deviates from the true value. Other DQ dimensions are required to describe the quality of sensor data streams. Further, the presented operator set is limited to the Cartesian product, selection, and projection, which does not suffice in common data stream applications.

Wang et al. [2001] focus on the accuracy of relational query results. Based on the accuracy of the input relation(s), they calculate the resulting accuracy of the selection, projection, and union operator. First, they do not discuss the join operator, which is of high importance in sensor environments, where multiple sensor streams have to be combined. Second, they analyze the data quality on the relation level, which corresponds to the complete data stream. The data quality would not be presented to the user until the complete sensor stream has been processed, which is far too late.

Scannapieco and Batini [2004] discuss the effects of the relational operators *union*, *intersection*, and *Cartesian product* on the data quality dimension completeness. In addition to the aforementioned restrictions, their calculation model only holds in the open-world assumption without missing values. Obviously, this algebra is not applicable to sensor data representing the closed-world scenario, where nonexisting values due to sensor failures cannot be avoided.

Klein [2007] presents the first attempt to analyze the data quality impact in data stream processing. The paper provides a data quality algebra for the timestamp-based join [Schmidt et al. 2005] using up- and downsampling for

Final edited form was published in "Journal of Data and Information Quality". 2009, 1(2), S. 1-28. ISSN 1936-1963.
https://doi.org/10.1145/1577840.1577845

Timestamp	...	210	220	230	240	250	260	270	280	290	300	310	320	330	340	350	360	370	380	390	400	...
Lifetime	...	300	298	295	292	292	292	292	283	274	265	255	252	250	242	233	206	195	190	187	184	...
Accuracy	...	3.0									3.3					2.78					2.86	...
Completeness	...	0.9									0.8					0.9					1	...

Fig. 1. Jumping DQ windows.

data stream rate adaptation as well as elementary numeric operators like algebraic operators, the threshold control, and plain aggregation functions. The focus lies on data completeness as well as on systematic and statistical numeric errors expressed by the data quality dimensions accuracy and *confidence*, respectively.

In this article, we extend the approaches of Klein [2007] to cover the expressiveness of the standard CQL (Continuous Query Language) presented in Arasu et al. [2003] including, for example, selection, projection, and sliding window aggregation. Moreover, we discuss operators derived from signal processing, which are frequently applied during technical data stream analysis. Moreover, this work extends the notion of data quality by discussing additional data quality dimensions, such as timeliness and data volume.

3. DATA QUALITY MANAGEMENT

A data stream D comprises a continuous stream of m tuples, consisting of n attribute values A_i $(1 \leq i \leq n)$ and the timestamp t. To allow for the efficient data quality management, the stream is partitioned into κ consecutive, nonoverlapping jumping data quality windows $\omega_i(k)$ $(1 \leq k \leq \kappa)$, each of which is identified by its starting point t_b, its end point t_e, the window size ω, and the corresponding attribute A_i. Beyond the sensor measurements $\upsilon_i(j)(t_b \leq j \leq t_e)$, the window contains a set of $|DQ|$ data quality information, each describing one DQ dimension $q \in DQ$, for example, the window accuracy $a_w(k)$ or the window completeness $c_w(k)$ as shown in Figure 1 with $\omega = 5$.

The generic data quality model allows for a variable number of data quality dimensions that are adaptable to various user requirements. To allow the comprehensive evaluation of sensor measurement streams, we propose a set of five data quality dimensions derived from the DQ categories provided by Wang and Strong [1996]. The intrinsic data quality dimension *accuracy* describes the maximal systematic numeric error of a sensor measurement. The *confidence* represents the maximal statistical error. While the intrinsic data quality dimensions characterize single data items, the contextual data quality refers to datasets. The *completeness* characterizes missing values in a dataset, while the *data volume* describes the amount of underlying raw data. The *timeliness* evaluates the temporal context of the data stream.

The window size ω can be defined independently for each stream attribute and/or window. Small jumping DQ windows result in high-granular data quality information at the expense of a higher data overhead. A wider window definition guarantees the important resource savings that are essential for data stream environments; this happens by risking DQ information with lower

Final edited form was published in "Journal of Data and Information Quality". 2009, 1(2), S. 1-28. ISSN 1936-1963.
https://doi.org/10.1145/1577840.1577845

granularity and decreased correctness due to error deviations introduced by the window-wise DQ aggregation. Therefore, we present the automatic data quality model control in Section 5, which optimizes the window size during the data stream querying by dynamically adapting it to the interestingness of the current data stream and/or data quality.

In the following, the determined data quality dimensions are defined and methods for data quality recording and jumping DQ window initialization are given.

3.1 Accuracy

The sensor accuracy describes the systematic measurement error resulting from static errors in the measurement process, for example, due to miscalibration, retroactions of the measuring method, or environmental influences on the measured values. This numeric absolute error is constant in sign and value.

Definition 1. The accuracy of a numeric measurement value defines the maximal, absolute, systematic error a, such that the real value $\hat{v}$ lies in the interval $[v-a; v+a]$ around the measured value v.

During the data quality recording, the window accuracy $a_w(k)$ is initialized with the help of the sensor's precision class given in the manufacturer's technical specification. For example, the precision class *5%* determines the absolute accuracy error of a pressure sensor with a maximum range of 60bar to a_w = 3bar.

3.2 Confidence

The confidence illustrates the statistical measurement error due to random environmental interferences (e.g., vibrations, shocks). Due to its random character, the statistical error scatters around the mean value μ. The confidence (interval) defines the bounds of this random distribution based on the variance σ^2 of the data items. Similar to the sensor's accuracy, the confidence is given as the maximal absolute error.

Definition 2. The confidence of a numeric data item is given as the statistical error ε defining the interval $[v-\varepsilon; v+\varepsilon]$ around the measurement value v containing the true value $\hat{v}$ with the confidence probability p.

Due to the statistical error distribution, the interval containing the true value $\hat{v}$ with p = 100% is of unlimited size. We set the confidence probability to 99%, which leads to an initial confidence interval of $\varepsilon = \sigma$. For example, the initial confidence of a data quality window including the pressure measurement values {18.7, 10.2, 21.7, 21.3, 19.8, 18.5, 20.2, 19.3, 19.8, 18.1} is set to ε_w = 4.1bar.

Aside from the recording of the initial statistical error, the confidence plays an important role during the data stream processing, where selection or sampling is applied to reduce the data volume. During selection, an uncertainty range due to limited accuracy and confidence is generated around the

Final edited form was published in "Journal of Data and Information Quality". 2009, 1(2), S. 1-28. ISSN 1936-1963.
https://doi.org/10.1145/1577840.1577845

threshold function, which may lead to falsely selected or unselected tuples, which distort results of operators applied in the following processing steps (see Section 4.3.1).

The information loss provoked by sampling represents statistical errors as well. For example, consider a data stream consisting of 10 values {1,1,1,1, 1,1,1,1,1,100} with the true average of 10.9 sampled with a sampling rate of 50%. If the sample contains the value 100, the average determined afterwards will be 20.8. Otherwise, the average based on the sample would be computed to 1. It is easy to see that these two results contain a nonnegligible deviation compared to the true average.

3.3 Completeness

The completeness addresses the problem of missing values due to sensor failures or malfunctions. Multiple strategies exist to deal with missing values in ETL processes and data cleansing [Lee et al. 1999]. In most cases the estimation or interpolation of missing values is aspired. The data quality dimension *completeness* helps to distinguish between measured data items and estimated or interpolated ones.

Definition 3. The completeness c is stated as the ratio of originally measured, noninterpolated values $\tilde{v}(j)(t_b \leq j \leq t_e)$ compared to the size of the analyzed stream partition.

To compute the window completeness $c_w(k)$, the analyzed stream part is given as the data quality window $w(k)$ of size ω. Based on the known stream rate, missing sensor values are counted. Given the knowledge of the window completeness, each data value in the respective window $[t_b, t_e]$ has the probability of $p = c_w(k)$ to be an originally measured data item instead of an interpolated one. Assume, for example, that the oil temperature sensor in a truck's hydraulic system briefly fails two times during a DQ window ($r = 1/\text{min}$, $\omega = 1\text{h}$). Then, the completeness for this window is set to $c_w = 0.967$.

3.4 Data Volume

The data volume describes the amount of raw data used to compute the result of a data stream (sub-)query. For example, the data volume defines the basis of an aggregation.

Definition 4. The data volume d defines the amount of raw data items $v_j (1 \leq j \leq d)$ used to derive the data item $v' = f(v_j)$.

The window data volume $d_w(k)$ is defined as the averaged data volume of the data items contained in the respective data quality window. Initially, the data volume is set to 1 for each data quality window because every data item accords to one measurement value.

3.5 Timeliness

There are two perceptions of the data quality dimension timeliness. On the one hand, timeliness can express the age of a specific data item as the difference

Final edited form was published in "Journal of Data and Information Quality". 2009, 1(2), S. 1-28. ISSN 1936-1963.
https://doi.org/10.1145/1577840.1577845

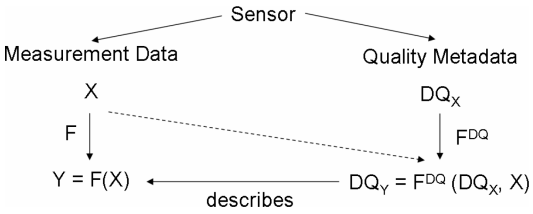


Fig. 2. Relations in data quality processing.

between the recording timestamp and the current system time. On the other hand, the timeliness can be interpreted as the punctuality of the data item with respect to the application context. The latter perception presumes the definition of the subjective application or user requirement and will not be regarded in this article.

Definition 5. The timeliness u defines the age of a data item $v(j)$ as the difference between the current system time and the timestamp of the data recording $t(j)$.

The timeliness takes an exceptional position. In contrast to other data quality dimensions, it can be calculated at runtime and must not be recorded, propagated, and processed during the data processing.

THEOREM 1. *The timeliness of a query result v' is defined as the maximal timeliness $u_v(j)$ of the underlying processed data items $v(j)$.*

Given these data quality definitions, we define the term data quality in the context of sensor data streams as follows.

Definition 6. The sensor data quality Q is described by the data quality dimensions accuracy a, confidence ε, completeness c, data volume d, and timeliness u.

The user is able to evaluate the data quality from three different points of view. Accuracy and confidence will be summarized at the end of the data processing to compute the overall numeric error of the stream processing result. Second, the completeness and data volume deliver insight into the quantity of the underlying raw data. Last but not least, the timeliness highlights the temporal context of data quality by giving the age of the query result.

4. DATA QUALITY PROCESSING

The correlation between sensor data and data quality processing is outlined in Figure 2. The sensor constitutes the source of the measurement data X and the data quality metadata DQ_X. During the data stream processing, Y is derived from raw data X by applying the complex function F composed of operators $o \in O$. The data quality function F^{DQ} is composed of the data quality operators $o^{DQ} \in O^{DQ}$ to compute the data quality DQ_Y, describing the derived knowledge $Y = F(X)$. The aim is to define a data quality operator o^{DQ} for each

Final edited form was published in "Journal of Data and Information Quality". 2009, 1(2), S. 1-28. ISSN 1936-1963.
https://doi.org/10.1145/1577840.1577845

Table I. Operator Classification

Operator Origin	Operator Type	Example	Data Manipulation				
			Modifying	Generating	Reducing		Merging
					Attribute	Item	
CQL	Projection				×		
	Selection					×	
	Join	Equi-Join	×				
	Aggregation	Slope Calculation					×
Signal Analysis	Sampling	Simple Random Sampling				×	
	Interpolation	Linear Interpolation		×			
	Spectral Analysis	Fourier Transformation					×

data stream operator o to extend the function F by a function F^{DQ} to define F' supporting the data stream as well as data quality processing.

In this article, we will present the operator extensions o^{DQ} for the DQ dimensions *accuracy*, *confidence*, *completeness*, and *data volume* introduced in Section 3. We will show that the data quality DQ_Y does not only depend on the data quality DQ_X. Many DQ operators also comprise the raw data X itself, as indicated by the dotted arrow in Figure 2.

In real-world applications, numerous operators are applied to data streams. First of all, there is the operator collection of the Continuous Query Language (CQL) [Arasu et al. 2003] providing query functionalities known from standard SQL, such as projection, selection, join, and aggregation. To adapt to the specific resource limitations during data stream processing, CQL supports the window-wise join and aggregation execution. Another important operator class is derived from the signal processing domain, for instance, sampling and frequency analysis. The list is completed by elementary numeric operations like addition or multiplication and the threshold control, which have been studied in detail in Klein [2007]. The operators discussed in this article are summarized in Table I. Where several operator implementations are possible, an example is provided for detailed analysis.

To analyze the influence on data quality, we evaluated the data stream operators in the context of data manipulation. Four operator classes can be distinguished: *data-modifying*, *data-generating*, *data-reducing*, and *data-merging* operators.

Data-modifying operators manipulate a data stream, while the stream rate stays constant. During data generation new data items are inserted into the data stream increasing the stream rate, whereas items are removed by data-reducing stream operators. The deletion of a complete attribute stream (projection) and deleting operations on specific data items (e.g., selection) have to be distinguished. Data-merging operators reduce the data volume by computing a compressed description of a dataset.

4.1 Data-Modifying Operators

Data-modifying operators have no effect on the data volume. The data quality dimension *completeness* is constant as well because data items are neither

Final edited form was published in "Journal of Data and Information Quality". 2009, 1(2), S. 1-28. ISSN 1936-1963.
https://doi.org/10.1145/1577840.1577845

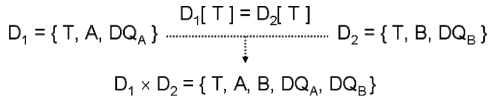


Fig. 3. Elementary join of equal timestamps.

deleted nor generated. Members of the class of data-modifying operators are unary algebraic operators like the square root or the threshold control. The filtering allows for the modification of signal-inherent frequency bands; for example, the low-pass filter is used to prevent aliasing due to data stream sampling. The data stream join constitutes another data-modifying operator, where two data streams are combined by copying both measurement streams as well as data quality information.

While the first two operator types are discussed in detail in Klein [2007], this section focuses on the data quality impact of the timestamp-based join and illustrates the handling of jumping data quality windows during the window-wise data stream join execution.

4.1.1 *Join of Synchronic Streams.* The simplest join approach assumes synchronous sensor data streams and builds one-to-one tuple pairs based on identical timestamps, as shown in Figure 3. Equal data stream rates do not suffice for this approach. The sensor data could be measured shifted against each other, so that no identical timestamps exist.

During the join of two data streams D_1 and D_2, data quality information DQ_A and DQ_B are not affected but copied to the resulting data stream, analog to the attribute sets A and B. Since the quality metamodel allows for attribute-independent DQ window sizes, different window size instantiations in incoming streams do not lead to problems during the join.

THEOREM 2. *The timestamp-based join of synchronous data streams has no impact on the data quality information, such that* $o^{DQ} : q'_w(k) = q_w(k)$ *for* $DQ = \{accuracy, confidence, completeness, data\ volume\}$ *and* $q = \{a, \varepsilon, c, d\}$, *respectively.*

4.1.2 *Timestamp-Join of Asynchronous Streams.* The assumption of data stream synchrony does not hold for typical application scenarios. Schmidt et al. [2005] present source-aware join strategies to join asynchronous data streams with different stream rates. The basic idea is to use sampling and interpolation techniques to adapt the stream rates and overcome phase shifts in the data streams.

The complex operator has to be split up as shown in Figure 4 to allow the tracking of the data quality impact. The data streams D_1 and D_2 are sampled and/or interpolated to be joined afterwards with the help of the timestamp-based, synchronous join approach. To quantify the influence of the overall join operator, the data quality influences of the basic operators *interpolation* and *sampling* have to be studied as illustrated in Sections 4.2 and 4.3.2. In this article, we focus on the timestamp-based joining of two sensor data streams.

Final edited form was published in "Journal of Data and Information Quality". 2009, 1(2), S. 1-28. ISSN 1936-1963.
https://doi.org/10.1145/1577840.1577845

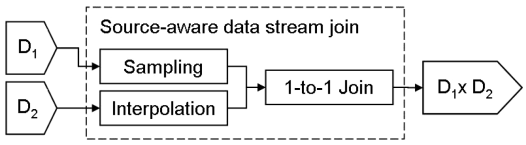


Fig. 4. Joining asynchronous data streams.

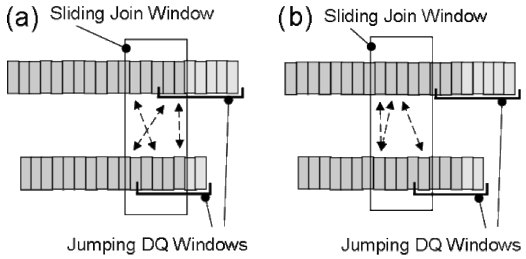


Fig. 5. Sliding window join.

However, the presented approaches can be easily extended to other ordered join attributes, where sampling and interpolation techniques are applicable.

4.1.3 *Window Join of Unordered Data Streams.* If the join attributes are not ordered, the window-wise data stream join is recommended to comply with restricted memory and CPU resource constraints in data stream environments. Comprehensive work has been done in the field of window joins [Moon 2006] and sequence matching [Kang et al. 2002]. While a sliding window join of two data streams is executed not all streaming tuples find join partners, independent from the specific join implementation. Thus, the window-wise join of unordered data streams includes an implicit sampling on one or both affected data streams. To track the influence of this sampling on the data quality dimension *confidence*, the implicit sampling rate has to be recorded for each jumping DQ window, while it overlaps with the sliding join window (Figure 5(a)). As soon as the sliding join has left the jumping DQ window (Figure 5(b)), the confidence can be updated as declared in Section 4.3.2 and the data quality can be propagated to the next operator in the processed query.

4.2 Data-Generating Operators

The second class of stream operators is given by data-generating operators. Data items are inserted into the data stream based on existing sensor data. The data generation increasing the data rate has to be tracked by updating the DQ dimension *completeness*, which represents the ratio of original measured data values. The generation factor r_g describes the multiplication factor of the stream length. For example, the data generation may be executed with the help of linear interpolation with $r_g = 2$.

Both accuracy and confidence have to be retrieved from existing DQ information, while data items are computed based on existing measurements. The same data generation strategy has to be applied to the DQ information as to the original sensor data. After the generation of data items, each DQ window is increased by the factor r_g. To achieve the goal of constant window size

Final edited form was published in "Journal of Data and Information Quality". 2009, 1(2), S. 1-28. ISSN 1936-1963.
https://doi.org/10.1145/1577840.1577845

during all processing steps, each DQ window has to be partitioned in r_g child windows. Concerning the data quality calculation, the first $r_g - 1$ windows, which inherit the DQ dimensions *accuracy*, *confidence*, and *data volume* from the parent window, differ from the last child window, where the data quality information for $DQ = \{accuracy, confidence, data\ volume\}$ is generated (e.g., interpolated) based on the respective parent window $w(k)$ and the subsequent window $w(k+1)$.

THEOREM 3. *During data generation, the window completeness c_w is divided by the generation factor r_g for each DQ window, such that $o^{completeness}: c'_w = c_w / r_g$. The data quality $q = \{a, \varepsilon, d\}$ is derived from the former data quality information for the first $r_g - 1$ windows ($k < j \cdot r_g$) and interpolated for the last window $w(k)$, where $k = j \cdot r_g$, such that*

$$o^{DQ} : q'_w(k) = \begin{cases} q_w(k) & k < j \cdot r_g (1 \leq j \leq \omega) \\ (\omega - 1) \cdot q_w(k) + \dfrac{q_w(k) + q_w(k+1)}{2} & k = j \cdot r_g (1 \leq j \leq \omega) \end{cases}.$$

4.3 Data-Reducing Operators

In contrast to data generation, data-reducing operators decrease the volume of the data stream to meet resource constraints like limited communication capability, restricted memory capacity, and processing power. In the following, the CQL operators *projection* and *selection* as well as the signal processing operator *sampling* are discussed.

The projection operator performs an attribute restriction of the data stream. While the original projection operator o can be seen as an attribute-wise selection on sensor data, the corresponding data quality operator o^{DQ} extracts the relevant DQ information.

4.3.1 *Selection.* During the selection, data items are extracted for further processing based on the constraint evaluation of a certain measurement attribute. Tuples that do not satisfy the selection criterion are discarded from the data stream.

The condition evaluation as the first step of the selection resembles the threshold control introduced in Klein [2007]. Here, the incoming data stream is evaluated against a given threshold resulting either in the Boolean *true* if the threshold holds or *false* for a threshold exceeding. The accuracy and confidence of the measurement value (a_v, ε_v) as well as the threshold function (a_b, ε_b) define the uncertainty range $\delta = a_b + a_v + \varepsilon_b + \varepsilon_v$ around the threshold function b. In the context of selection, this approach reveals the following problems for data items lying in the uncertain range δ.

(1) Sensor measurements in the uncertain range are selected, even though the true value may not exceed the threshold constraint.
(2) Data items are not selected, although the selection condition may be met by the true value.

The false positives and false negatives may balance if there is a uniform data distribution in the uncertain range. Otherwise—which is far more likely—the

Final edited form was published in "Journal of Data and Information Quality". 2009, 1(2), S. 1-28. ISSN 1936-1963.
https://doi.org/10.1145/1577840.1577845

false selection leads to erroneous results if aggregation operators are applied during further data processing. The aggregated value is either too high because too many data items have been selected, or to low because relevant data items are missing. The DQ dimension *confidence* was introduced in Section 3.2 to monitor this type of statistical errors.

THEOREM 4. *The statistical error introduced by the faulty selection of data items in the uncertain range is summarized in the data quality dimension confidence. The new window confidence* $\varepsilon'_w(k)$ *is calculated as the root mean square of the former* $\varepsilon_v(k)$ *plus the new statistical error* $\varepsilon_{w,new}(k)$*, where* $\sigma^2(w)$ *is the variance and* r_{sel} *constitutes the selection rate for the corresponding data quality window. The parameter d describes the confidence probability p as the* $(p-1)/2$*-quantile [Haas 1997].*

$$o^{confidence} : \varepsilon'_w(k) = \sqrt{\frac{1}{\omega}\sum \varepsilon_v(k)^2 + \varepsilon_{w,new}(k)^2}$$
$$\varepsilon_{w,new}(k) = \frac{d\cdot\sigma(w)}{\sqrt{\omega\cdot r_{sel}}} \cdot \sqrt{1-r_{sel}}$$

Assuming that the selection criterion is independent of the data quality dimension *completeness*, the selection probability of an original measurement equals the selection likelihood of an interpolated value. Then, the selection operator has no direct influence on the completeness of a data stream. The same holds for the data quality dimensions *accuracy* and *data volume*.

Otherwise, the correlation of the selection attribute and data quality information can be measured with the help of the Pearson correlation coefficient. For example, extreme environmental parameters like high temperature or pressure may lead to an increase of sensor failures or systematic errors and decrease the completeness or accuracy, respectively.

If there is a positive correlation, high data values of the selection attribute correspond to a high quality (e.g., high completeness or high accuracy) of the respective data quality window. If the selection operator shifts the average data distribution to higher measurement values, the quality is likely to increase. We propose using the normalized alteration $\Delta\mu$ of the mean value of the selection attribute in the respective DQ window as indicator for the quality improvement or decrease for positive or negative correlations, respectively. In both cases, the selected data items build up new windows, where the window data quality resides in the average of the data items' former quality values $q'_v(j)$.

THEOREM 5. *The selection has no influence on the data quality DQ = {accuracy, completeness, data volume} if the selection attribute and data quality information are not correlated. Otherwise, the quality shift depends on the normalized alteration of the selection attribute values and the correlation coefficient cc, such that*

$$o^{DQ} : q'_w(k) = \frac{1}{\omega}\sum q'_v(j) \qquad (\omega(k-1) < j \leq \omega k)$$
$$q'_v(j) = q_w(k)\cdot(1+cc\cdot\Delta\mu)$$
$$\Delta\mu = (\mu_{aposteriori} - \mu_{apriori})/\max(\mu_{apriori}, \mu_{aposteriori})$$

4.3.2 *Sampling.* The sampling operator reduces the data stream volume. A given amount of data items is randomly skipped. To allow the correct

Final edited form was published in "Journal of Data and Information Quality". 2009, 1(2), S. 1-28. ISSN 1936-1963.
https://doi.org/10.1145/1577840.1577845

reconstruction of the original signal from the sample, aliasing effects have to be prevented. Therefore, the usage and applicability of the low-pass filter, which eliminates high frequencies in the signal stream, is shown in Klein [2007]. There exist multiple strategies to optimize the data stream sampling [Al-Kateb et al. 2007; Dash and Ng 2006]. The approach presented here holds for every sampling method that creates a simple random sample. The sampling rate r_s defines the resulting stream rate $r' = r_s r$ based on the former stream rate r.

The sampling can be regarded as a selection with a random selection criterion of high flexibility. Hence, Theorem 5 can be applied to calculate the statistical error as the new window confidence $\varepsilon_w(k)$ by replacing the selection rate r_{sel} by the sampling rate r_s.

In contrast to the selection, the sampling operator does not influence the completeness of a data stream. The simple random sample is independent of any data stream (quality) characteristics. If each data item is sampled with equal probability r_s, the fraction of originally measured or interpolated data values does not change. However, new DQ windows are built up by averaging the incoming data quality values. Theorem 6 can be used to compute the resulting data quality information $DQ = \{accuracy, completeness, data\ volume\}$ by setting the correlation coefficient $cc = 0$.

4.4 Data-Merging Operators

Data-merging operators aggregate a given set of data items to reduce the data volume and/or extract complex knowledge. The merging operators compress the incoming data to one output value or create a synopsis consisting of several data items. First, we present the calculation of window completeness and data volume for all data-merging operators in a generic way. Then, we detail the accuracy and confidence computation for the aggregation and spectral analysis. The merging operators of binary algebra are discussed in Klein [2007]. At the end of this section, the relation between jumping data quality windows and data merging in sliding windows is discussed.

The aggregated window completeness c_w is independent of the applied merging function. However, it depends on the size of the created synopsis.

THEOREM 6. *The aggregated window completeness $c_w(k)$ is defined as follows.*

$$o^{completeness} : c_w(k) = \underbrace{\frac{1}{\omega} \sum_{h=\omega(k-1)+1}^{k\omega} \overbrace{\frac{n}{l} \sum_{j=l/n(h-1)+1}^{l/n \cdot h} c_v(j)}^{c'_v(j)\ \ldots\ compl.\ of\ one\ synopsis\ entry}}_{average\ of\ synopses\ building\ one\ DQ\ window}$$

In a synopsis of size n built upon l raw data items, each synopsis entry represents l/n underlying measurement values. The completeness $c'_v(j)$ of one synopsis item $v'(j)$ is defined as the average of the completeness $c_v(j)$ describing the incoming tuples, which is again averaged to compute the completeness $c_w(k)$ of the newly built aggregate result windows $w(k)$.

Final edited form was published in "Journal of Data and Information Quality". 2009, 1(2), S. 1-28. ISSN 1936-1963.
https://doi.org/10.1145/1577840.1577845

Similar to the completeness, the aggregated window data volume d_w does not depend on the aggregation type but rather on the size of the created synopsis. The data volume $d'_v(j)$ of each synopsis item $v'(j)$ summarizes the data capacities of l/n input items. Similar to the aggregated window completeness, the data volumes of the included synopsis items are averaged to calculate the window data volume $d_w(k)$.

THEOREM 7. *The data volume of an aggregated data quality window consists of the averaged sum of the measurement items' data volume values underlying the synopsis entries included in the respective window.*

$$o^{datavolume} : d_w(k) = \underbrace{\frac{1}{\omega} \sum_{h=\omega(k-1)+1}^{k\omega} \overbrace{\sum_{j=l/n(h-1)+1}^{l/n \cdot h} d_v(j)}^{d'_v(j) \ldots vol.\ of\ one\ synopsis\ entry}}_{average\ of\ synopses\ building\ one\ DQ\ window}$$

In contrast to the completeness and data volume analysis, during the examination of accuracy and confidence, it has to be distinguished between the different operator types, like aggregation or spectral analysis.

4.4.1 *Aggregation.* In this paragraph, we first introduce the aggregated window data quality calculation in general and then show the calculation steps reflecting a suitable example. The aggregation is commonly executed in combination with data grouping, allowing time-based window aggregations. The grouping based on the timestamp can be applied to build groups representing a given time interval (e.g., 10min) or consisting of a specific number of data tuples (e.g., 100 tuples). Thus, the grouping divides the data stream in g intervals with either varying length l_i depending on the respective stream rate or fixed length $l = m/g$ that is equal for all intervals.

Aside from the timestamp-based aggregation, every other sensor stream attribute may serve as grouping attribute leading to groups of varying length. The data quality aggregation methods presented here and in Klein [2007] can be applied without modifications to any grouping attribute.

During aggregation, each group of data items is summarized to compute a single data result, the *aggregate*. This data value represents not only a certain point in time but a time interval. The timestamp has to be adjusted to the form $[t_b, t_e]$ to represent this fact. The timeframe defining the grouping for an aggregation operator is independent from the window size ω for data quality calculation.

For the DQ aggregation, two steps have to be distinguished. First, the data quality of one *aggregate* is calculated based on all incoming tuples' DQ information. Second, the resulting aggregates are bundled to form new windows of size ω.

As an example, we will discuss the slope calculation, which is reused in Section 5 for the data quality model control. To calculate the average measurement slope in a given timeframe, the measurement stream is approximated

Final edited form was published in "Journal of Data and Information Quality". 2009, 1(2), S. 1-28. ISSN 1936-1963.
https://doi.org/10.1145/1577840.1577845

with the help of a linear fitting $y=mx+a$. The regression algorithm of Least Squares [Papula 2006] is applied.

$$m = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

To calculate the accuracy and confidence of one resulting aggregate, respectively, the incoming value accuracies or confidences are summed up (squared) weighted with the partial derivative $\partial m/\partial y_i$ following the Gaussian error propagation. To build new data quality windows, the aggregate data quality is averaged.

$$o^{accuracy} : a'_v(j) = \sum_{j=1}^{l} \left|\frac{\partial m}{\partial y_j}\right| a_v(j)$$

$$o^{confidence} : \varepsilon'_v(j) = \sqrt{\sum_{j=1}^{l} \left(\frac{\partial m}{\partial y_j}\right)^2 \varepsilon_v(j)^2}$$

THEOREM 8. *For the slope calculation, the window accuracy $a_w(k)$ and confidence $\varepsilon_w(k)$ are defined as the (squared) average of the aggregate accuracy $a_v(j)$ and confidence $\varepsilon_v(j)$, respectively.*

$$o^{accuracy} : a'_w(k) = \frac{1}{w} \sum_{j=w(k-1)+1}^{wk} a'_v(j)$$

$$o^{confidence} : \varepsilon'_w(k) = \frac{1}{w} \sqrt{\sum_{j=w(k-1)+1}^{wk} \varepsilon'_v(j)^2}$$

4.4.2 *Spectral Analysis.* As a second data-merging operator, the spectral analysis surveys the signal's frequency spectrum. The amplitudes and phases of the signal-inherent frequency bands are delivered by transforming time to frequency domain. The frequency spectrum can be determined with the help of the Fourier analysis as well as with wavelet transformations.

The relation between the time and frequency domain of periodic signals is stated in the Fourier transformation shown in the following.

$$X_n = \frac{1}{T} \int_T x(t) \cdot e^{-jn\omega_0 t} dt$$

Assuming that the transformation is executed separately for every data quality window, the time series $x(t)$ is composed of the true measurement value $\hat{x}(t)$ and the absolute error $\Delta x = a_w + \varepsilon_w$. Hence, the Fourier transformation is split into two signal transformations.

$$X_n = \underbrace{\frac{1}{T} \int_T \hat{x}(t) \cdot e^{-jn\omega_0 t} dt}_{\hat{X}_n} + \underbrace{\frac{1}{T} \int_T \Delta x \cdot e^{-jn\omega_0 t} dt}_{\Delta X_n = 0}$$

The first part represents the transformation of the true streaming signal $\hat{X}_n$. The second part describes the Fourier transformation of Δx. The Fourier transformation of the constant Δx is 0. Thus, the systematic and statistical errors

Final edited form was published in "Journal of Data and Information Quality". 2009, 1(2), S. 1-28. ISSN 1936-1963.
https://doi.org/10.1145/1577840.1577845

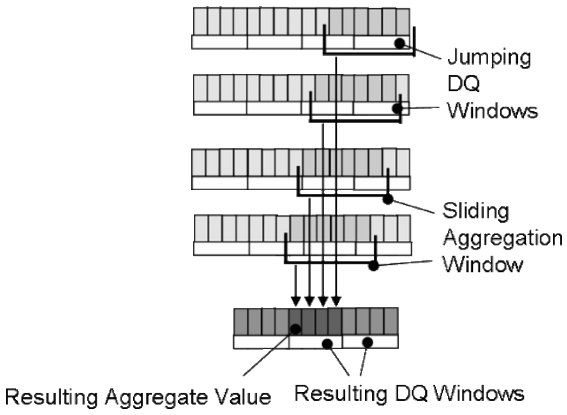


Fig. 6. Sliding window aggregation.

in the time domain represented by $a_w + \varepsilon_w$ are not visible in the frequency domain. By separately analyzing the signal-inherent frequencies for each data quality window, the frequency bands can be determined without deviation.

4.4.3 *Sliding Window Merging.* There are numerous possibilities of sliding window aggregation. On the one hand, there are landscape windows with a constant starting point that grow with time, like the turnover sum of a shop continually calculated beginning each morning at shop opening. On the other hand, sliding windows of constant size can be defined by their step size and window length. For example, a manager could be interested in the average price of a stock fund over the past hour (window length l), updated every minute (step size s). Figure 6 illustrates the sliding window aggregation for $l = 6, s = 1, \omega = 4$. Independent from the size of the sliding aggregation window, the resulting aggregation values form new data quality windows.

The sliding window aggregation does not interfere with the jumping data quality window computation. The same strategies apply to calculate window accuracy and confidence as well as data volume and completeness. Compared to standard aggregation in fixed groups, the number of resulting data items varies. The resulting data quality window does not represent the $l\omega$ underlying data items as during standard group-by but the smaller set of $l + s(\omega - 1)$ items due to the overlapping grouping.

5. DATA QUALITY MODEL CONTROL

In Section 3, the definition of the data quality window size was introduced as a challenging task. A trade-off has to be found to meet the competing requirements of low data overhead for DQ propagation in data streaming environments and fine-granular data quality information. Due to the data quality aggregation in jumping windows, the DQ information can only be streamed at the end of each respective window. The longer the window is defined, the later the DQ characteristics are available at the end application.

Final edited form was published in "Journal of Data and Information Quality". 2009, 1(2), S. 1-28. ISSN 1936-1963.
https://doi.org/10.1145/1577840.1577845

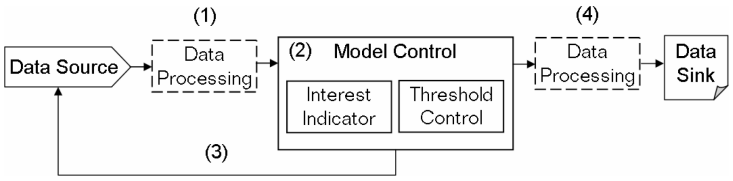


Fig. 7. Data quality model control.

In the following, we present a framework computing the window size based on the *interestingness* of data stream characteristics. We propose two algorithms for different use cases. The interestingness is not a data quality dimension such as accuracy or completeness. It does not describe the quality considering the goodness or excellence of data, but rather characterizes the data stream in the context of a specific application scenario.

Ranges of high interest, for example, data peaks, high fluctuations, or a threshold exceeding, are controlled with fine-granular data quality windows. For stream parts of low interest, such as constant sensor measurements, wider DQ windows are sufficient. Further, the window size configuration based on the DQ information itself is possible. If the data quality changes frequently, only small DQ windows provide suitable information. For mainly constant data quality low-granular DQ windows can be defined.

Figure 7 presents the data quality model control, which reuses operators described in Section 4 for the window size adaptation. The data processing starts with a fixed window size (1), defined by the model user. The control operator (2) may be applied anywhere in the data processing graph. It consists of a functional operator indicating the interestingness of the current data stream or data quality information and a threshold control, which evaluates the current interest indicator. If the threshold is exceeded, the window size has to be decreased. A notification is transferred back to the data source(s) (3). The DQ window size is adapted, so that the data stream provides data quality information with finer granularity (4). To support different levels of interestingness, more than one threshold can be applied, resulting in a set of interest classes.

The source notification is required because fine-granular DQ information cannot be retrieved from low-granular data quality. The window size cannot be adapted immediately but deferred to the detection of changing interestingness. To minimize the delay, the control operator has to be applied as soon as possible in the quality propagation graph.

The generic algorithm for window size adaptation is illustrated in Listing 1. It presents the framework for various application scenarios, which can be customized by choosing appropriate specifications for the functions interestingness(), decreaseDQSize(), and increaseDQSize().

```
WHILE(DS.hasNext())
    IF interestingness(DS) $>$ threshold
    THEN DO decreaseDQSize()
    ELSE DO increaseDQSize()
END WHILE
```

Listing 1. Framework for DQ model control.

Final edited form was published in "Journal of Data and Information Quality". 2009, 1(2), S. 1-28. ISSN 1936-1963.
https://doi.org/10.1145/1577840.1577845

Table II. Interest Indicators and Operators

	Interest Indicator	**DQ Control Operator Pattern**
currentValue()	Extraordinary value ranges	→ Threshold Control →
slidingSlope()	Extraordinary value alterations	→ Slope Calculation → Threshold Control →
fft()	Unsteadiness	→ FFT → Threshold Control →
fftSlope()	Changing periodicity	→ FFT → Slope Calculation → Threshold Control →

Table II shows exemplary definitions of the interestingness function together with respective candidate operators used during the DQ model control.

—*currentValue()*. Extraordinary measurement values, for instance, very high temperatures in a critical range, can be detected with the help of the threshold operator, which in this case also takes over the role of the interest indicator.

—*slidingSlope()*. Extraordinary value alterations like the fast rising of a measurement value can be perceived with the help of a sliding slope aggregation over the respective data quality. For example, during the oil monitoring, the pressure rising in a hydraulic piston is monitored to guarantee the early detection of a pressure loss due to sealing wear-out. During normal operation, the DQ windows are defined rather large to spare restricted communication capacities resulting in minor quality estimation. As soon as a critical threshold of pressure rising is exceeded, the DQ windows are increased to enable the clear detection of the oil pressure state with fine-granular DQ information.

—*fft()*. The unsteadiness of measurement values is the third indicator for important stream partitions. The spectral analysis, for example, with the Fast Fourier Transformation (FFT), detects signal-inherent high frequencies, whose amplitudes may be evaluated with the help of the threshold control.

—*fftSlope()*. Another frequency-based model control evaluates the alteration of the data stream's frequency spectrum. After the FFT has transformed the signal from time to frequency domain, the slope calculation together with the threshold control recovers significant shifts. An example is a complex manufacturing line, whose status is monitored in a machine-specific characteristic curve. Due to constantly repeated manufacturing processes, the curve exhibits characteristic periods. Again, the size of the DQ window will be large during the normal state. However, deviations from the usual process run, which break the known periods, are a strong indication of a possible machine malfunction. Now, the DQ windows have to be increased to clearly identify the source of the problem.

Final edited form was published in "Journal of Data and Information Quality". 2009, 1(2), S. 1-28. ISSN 1936-1963.
https://doi.org/10.1145/1577840.1577845

In the following, two example algorithms implementing the DQ model control framework are presented. They are evaluated in Section 6.6. Listing 2 shows the algorithm for the Threshold Size Control (TSC). While the current measurement value stays below the threshold, the window size is halved until the minimal size is reached. During the threshold exceeding, the window size is doubled up to the maximal window size.

```
WHILE(DS.hasNext())
    IF currentValue(DS) $>$ threshold
    THEN DO size = max(minSize, size / 2)
    ELSE DO size = min(maxSize, size * 2)
END WHILE
```

Listing 2. TSC – Threshold Size Control.

The Slope Size Control (SSC) is presented in Listing 3. Here, the interestingness of the data stream is given as the degree of measurement rising. The slope is aggregated in sliding windows of two times the current DQ window size.

```
WHILE(DS.hasNext())
    IF slidingSlope(DS, 2*size) $>$ threshold
    THEN DO size = max(minSize, size / 2)
    ELSE DO size = min(maxSize, size * 2)
END WHILE
```

Listing 3. SSC – Slope Size Control.

The longer the threshold is exceeded, the more interesting is the current data stream part. Therefore, the window size is increased as long as the streaming values stay above the threshold.

The algorithms TSC and SSC update the data quality window size with the factor 2. More elaborate size adaptations are possible. The window size may directly depend on the magnitude of the threshold exceeding, such that $\omega' = f(interestingness(DS), threshold)$. The specific definition of the function f depends on expert knowledge in the respective application domain and will not be discussed here.

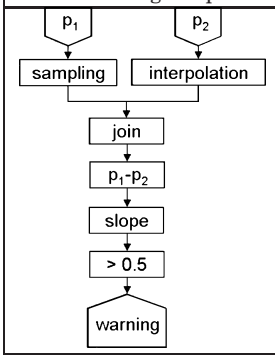
The proposed framework enables the automatic control of the data quality processing. It provides a trade-off between the efficient data quality propagation and the support of high-granular DQ information for interesting data stream partitions.

6. EVALUATIONS

The evaluation of the proposed definitions and concepts for data quality processing is based on a practical application scenario. In a hydraulic cylinder, the pressure loss is monitored with the help of two pressure sensors p_1 and p_2. To calculate the pressure difference, the sensor streams have to be synchronized with sampling and interpolation operations. A warning is raised when the increasing slope of the pressure loss exceeds 0.5bar per hour. Table III shows the processing graph and the respective evaluation tasks.

Final edited form was published in "Journal of Data and Information Quality". 2009, 1(2), S. 1-28. ISSN 1936-1963.
https://doi.org/10.1145/1577840.1577845

Table III. Evaluation Strategy

Processing Graph	Evaluated Operator	Section
	Sampling	6.1
	Interpolation	6.5
	Join and Binary Operators	6.3
	Slope Aggregation	6.4
	Selection	6.2

In order to validate the postulated theorems, we extended the data stream system PIPES [Kraemer and Seeger 2004] developed by the Research Group Database Systems at Philipps-University Marburg. We integrated functionalities to instantiate and manage different data quality dimensions. Furthermore, we implemented the data quality processing operators as defined in Section 4.

To show the benefit and practicability of our approach, we generated artificial data streams simulating true data $\hat{X}$ and added noise to mirror statistical and systematic measurement errors. Moreover, we simulated sensor failures by randomly skipping data items with a given failure probability. To evaluate the data quality estimation based on the proposed data quality operators $o_i \in F^{DQ}$, we executed the data processing F for both the true data streams as well as the noisy streams representing real-world sensor measurements X. Finally, we compared the true error provided in the difference between Y and $\hat{Y}$ with the quality estimation DQ_Y, as shown in Figure 8.

The data quality dimensions accuracy and confidence represent numerical measurement errors. They are validated together with the help of the relative error deviation *rel_dev* by comparing the average of the numerical difference of the true and noisy data streams with the average of the sum of estimated window accuracy and confidence over all data quality windows.

$$rel\_dev = \frac{\frac{1}{m} diff(Y,\hat{Y}) - \frac{1}{\kappa} \sum_k a_w(k) + \varepsilon_w(k)}{\frac{1}{m} diff(Y,\hat{Y})}$$

$$diff(Y,\hat{Y}) = |Y - \hat{Y}|$$

6.1 Sampling

Figure 9 shows the relative error deviation due to sampling with the given sampling rates and a subsequent application of a sum-aggregation with $l = 10$. The error deviation normalized by the sampling rate depends on the data quality window size. The wider the data quality window, the less precisely the measurement error can be predicted.

Final edited form was published in "Journal of Data and Information Quality". 2009, 1(2), S. 1-28. ISSN 1936-1963.
https://doi.org/10.1145/1577840.1577845

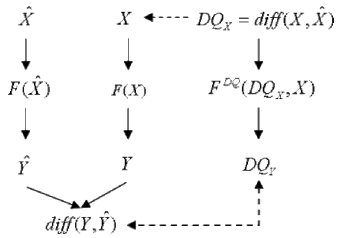


Fig. 8. Evaluation strategy.

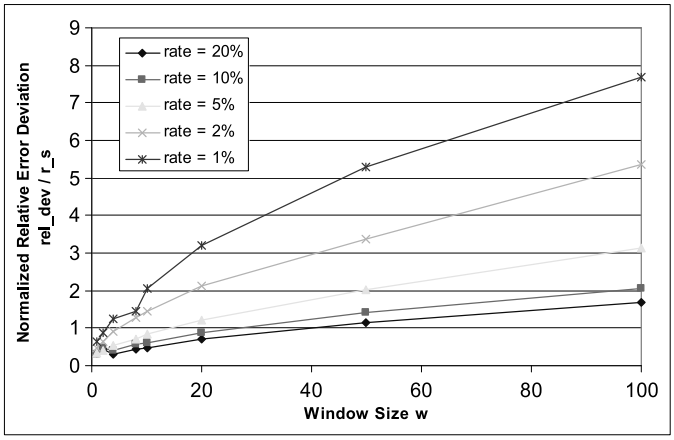


Fig. 9. Relative error due to sampling.

The sampling operator introduces a statistical error by deleting measurement items. For low sampling and selection rates, which remove many data items from the stream, the statistical error captured in the DQ dimension confidence dominates the overall numerical measurement error. For high sampling or selection rates, the systematic error caused by the sensor imprecision and captured in the dimension accuracy exceeds the statistical error.

6.2 Selection

Figure 10 shows the decreasing confidence due to selection with the threshold 50. While the true data stream remains below the given threshold, the noisy measurements exceed the selection bound (marked gray). Thus, data elements are falsely selected and removed from the stream, resulting in a data quality reduction. The absolute reduction value depends on the data quality window size. Small data quality windows ($\omega = 2, \omega = 4$) allow a fine-granular analysis of the selection result, so that data quality peaks (A) can be detected. When the confidence is aggregated over wider data quality windows, the resulting data quality is supposed to be lower (B).

The relative error deviation *rel_dev* due to selection and a subsequent application of the average aggregation ($l = 10$) is illustrated in Figure 11. The underlying data stream contained normally distributed data items $v_j \in [20, 180]$. For low selection rates ($v_j < 25$), the statistical error represented by the confidence is amplified and dominates the overall numerical error. As shown in

Final edited form was published in "Journal of Data and Information Quality". 2009, 1(2), S. 1-28. ISSN 1936-1963.
https://doi.org/10.1145/1577840.1577845

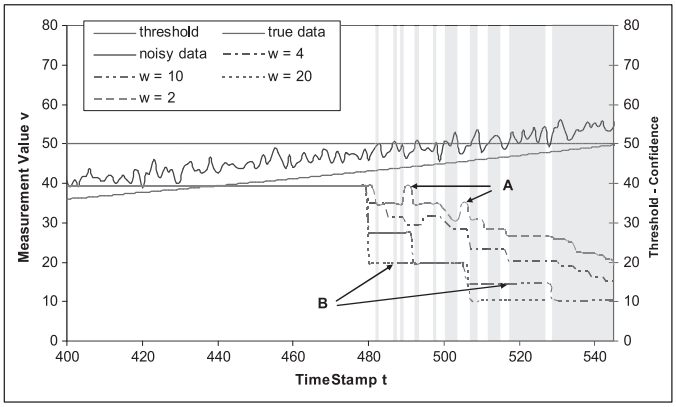


Fig. 10. Confidence decrease due to selection.

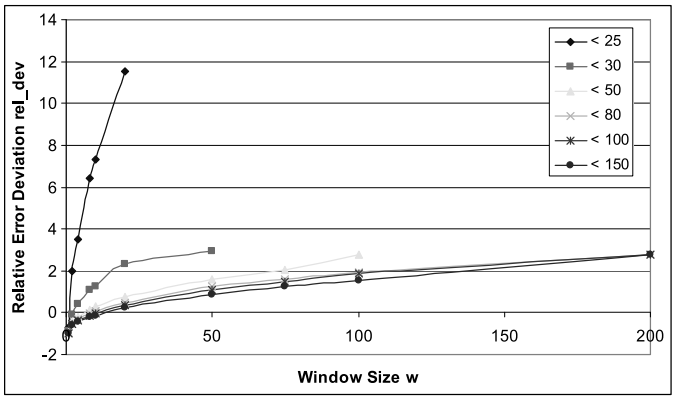


Fig. 11. Relative error due to selection.

Figure 10, the wider the DQ window, the less precisely the confidence can be estimated, so that the respective curve in Figure 11 rises sharply. On the other hand, for high selection rates ($v_j < 80, v_j < 150$), the systematic error represented by the accuracy bestrides the numerical error. Thus, the relative error deviation shows a lower slope.

6.3 Join and Binary Operators

Figure 12(a) illustrates the relative error deviation for binary algebraic operators $y = f(x_1, x_2)$. After executing a timestamp-based data stream join, an addition, subtraction, multiplication, or division is applied. The accuracy and confidence propagation for all binary operators is founded on the Gaussian deviation rules. Therefore, the relative error deviations are of equal order of magnitude. For small data quality windows, the estimated error deviates about 40%, while large DQ windows result in an error deviation of around 100%. In Figure 12(b), the completeness and data volume calculation is shown. The window completeness of y constitutes the average of $c_w(x_1)$ and $c_w(x_2)$, whereas the resulting data volume $d_w(y)$ constitutes the sum of incoming data volume values $d_w(x_1)$ and $d_w(x_2)$.

Final edited form was published in "Journal of Data and Information Quality". 2009, 1(2), S. 1-28. ISSN 1936-1963.
https://doi.org/10.1145/1577840.1577845

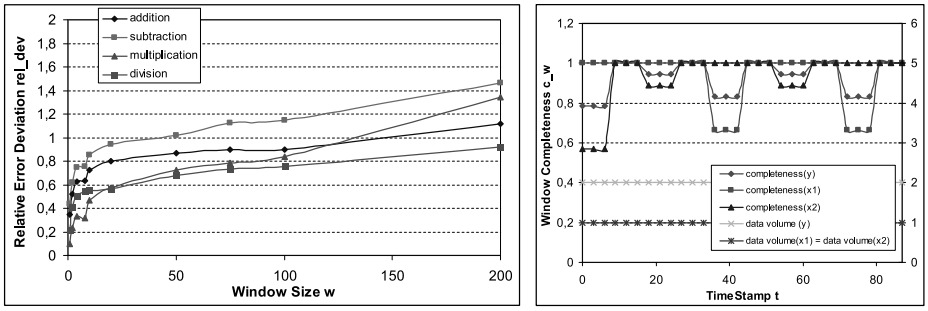


Fig. 12. Binary operators: (a) relative error deviation (b) completeness and data volume.

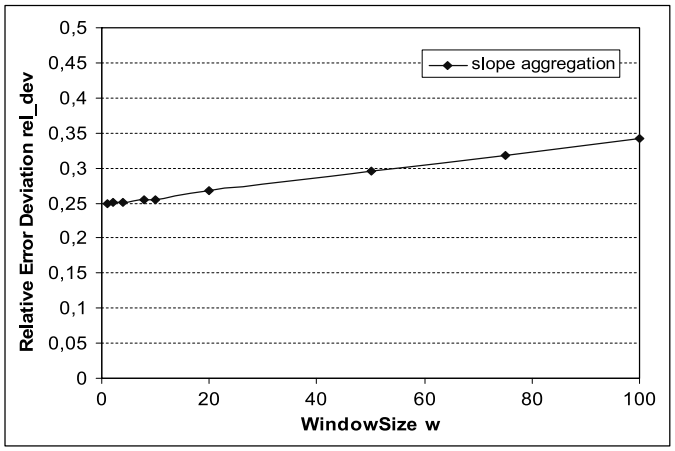


Fig. 13. Relative error deviation of slope aggregation.

6.4 Slope Aggregation

Aside from standard aggregations like average() and sum(), we presented the slope calculation as a useful tool for sensor stream monitoring. Figure 13 shows the linear rise of the relative error deviation *rel_dev* based on the window size ω, where the constant term 0.25 depends on the particular systematic error.

Figure 14 characterizes the window completeness after slope aggregation. The true completeness (gray) is averaged for each data quality window. Thus, fine-granular DQ windows allow a more detailed analysis of sensor failures.

6.5 Interpolation

During the interpolation of measurement errors with $r_g = 2$, the data quality is interpolated as well. Thus, the interpolated window accuracy (Figure 15, int) is nearly congruent with the prior window accuracy values (Figure 15, wo_int).

In contrast to the DQ dimensions *accuracy*, *confidence*, and *data volume*, the *completeness* is reduced by the interpolation factor.

Final edited form was published in "Journal of Data and Information Quality". 2009, 1(2), S. 1-28. ISSN 1936-1963.
https://doi.org/10.1145/1577840.1577845

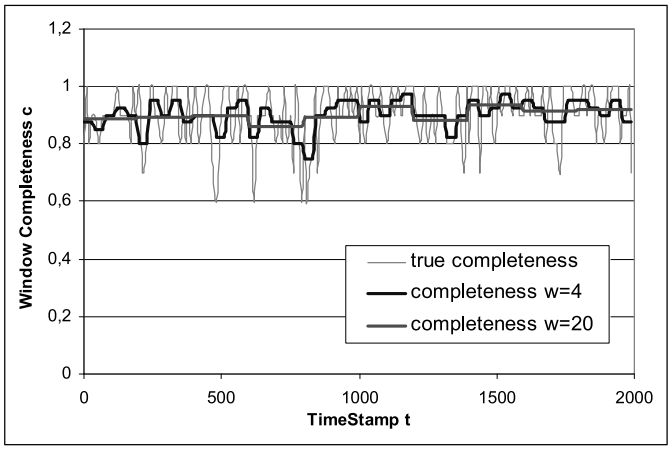


Fig. 14. Window completeness.

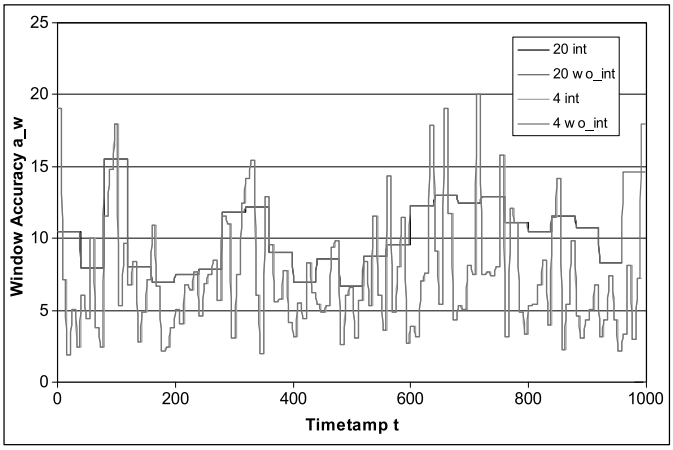


Fig. 15. Interpolation of window accuracy.

6.6 Model Control

In this section, the advantages of the automatic DQ window size adaptation are depicted. The previous evaluations showed that smaller DQ windows result in fine-granular data quality information which better estimate the true error. Thus, the automatic window size decrease for interesting data stream partitions results in better data quality information for these areas. However, smaller data quality windows increase the overhead for DQ transfer. The overall data stream rate r is defined as follows.

$$r = \sum_{i=1}^{n} \left(1 + \frac{q_i}{\omega_i}\right)$$

The attribute number is given as n, while q_i defines the number of propagated DQ dimensions for each attribute and ω_i states the window size. A compromise between the stream rate and the data quality granularity has to be found.

Final edited form was published in "Journal of Data and Information Quality". 2009, 1(2), S. 1-28. ISSN 1936-1963.
https://doi.org/10.1145/1577840.1577845

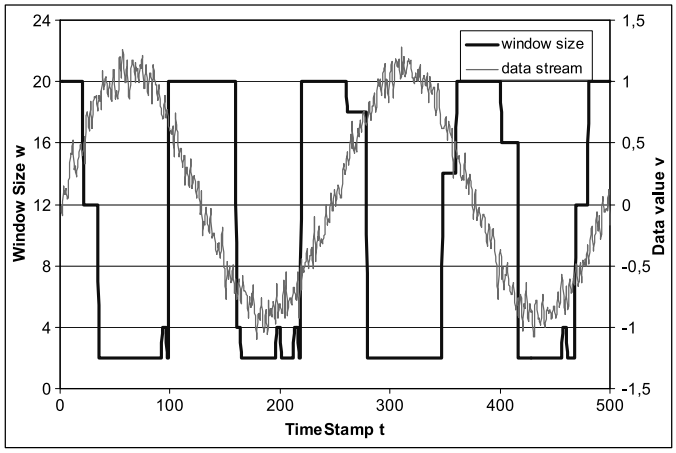


Fig. 16. Dynamic window size adaptation.

Therefore, the data quality management was extended to allow dynamic window size configurations. Moreover, the model controller was introduced as a novel data stream operator, regulating the DQ window size based on the interestingness of the streaming measurements. The maximal window size is 20; the minimal size is set to 2.

First, we evaluated the window size adaptation with the Threshold Size Control. Areas of interest are defined with the threshold $|v| > 0.7$. Figure 16 shows the data stream and adapted window sizes. When measurements enter an area of interest, the window size is reduced from 20 to 2. When the area is exited, the window size is increased. We tested the model controller without prior data processing steps (see Figure 7). There is no delay in the window size configuration. As soon as the interesting stream part is detected, the window size is decreased. When prior operators are applied, the delay corresponds to the processing time of these preceding operations. The window size fluctuation ($\omega = 2 \leftrightarrow \omega = 4$) result from data stream noise expressed in the DQ dimensions *accuracy* and *confidence*.

Figure 17 shows the window size adaptation with the Slope Size Control. Interesting data stream parts are defined with the slope threshold $|m| > 0.01$. Hence, there are small data quality windows when the measurements are rising or falling sharply. The window size adjustment is slightly deferred due to the more complex calculation of the sliding slope. Again, the delay will increase if prior data processing is executed and the fluctuation is caused by noise of the data stream.

7. CONCLUSIONS

In this article, we presented an efficient way to manage data quality in data streams. For a comprehensive evaluation of sensor measurements, we defined the data quality in the context of streaming data and proposed five data quality dimensions: accuracy, confidence, completeness, data volume, and timeliness.

To meet resource constraints in data stream environments, data processing is essential to reduce the streaming data volume. Operators retrieved from

Final edited form was published in "Journal of Data and Information Quality". 2009, 1(2), S. 1-28. ISSN 1936-1963.
https://doi.org/10.1145/1577840.1577845

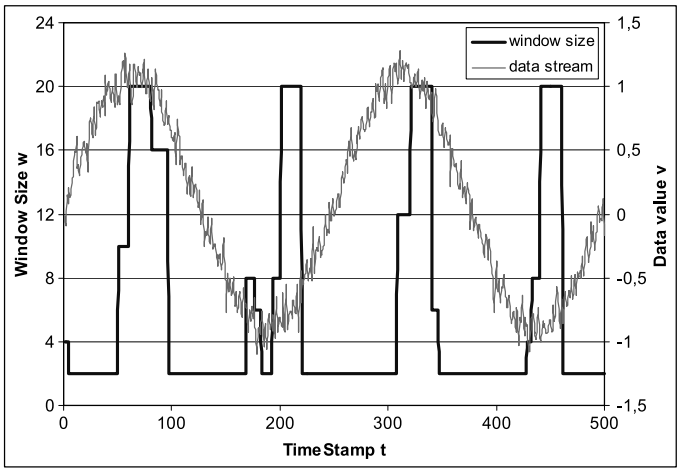


Fig. 17. Dynamic window size adaptation.

traditional data stream querying and the signal processing domain are applied to extract complex knowledge from raw data streams. We analyzed these operators to track their impact on the given data quality dimensions. Hence, an expansive data evaluation is enabled and the number of faulty business decisions is reduced.

Moreover, we proposed the reuse of data stream operators to allow for the automatic data quality model control. The DQ window size is adapted to the data stream interestingness. Fine-granular data quality information can be provided for interesting stream partitions by maintaining the overall efficient DQ transfer.

To show the practicability of the presented data quality processing theorems, we integrated the proposed DQ operators in the data stream system PIPES and compared the true error of generated random data streams with the calculated data quality estimation. Further, we demonstrated the benefits of the Threshold and Slope Size Control.

Future work will comprise the enhancement of operators presented in this article by complex operators derived from the research field of knowledge discovery, such as clustering, decision tree, and prediction algorithms. Another aspect is the complete implementation of the proposed DQ operators and data quality model control. The goal is to provide an end-to-end architecture ranging from sensors to their respective applications, which will allow a transparent data quality capturing, processing, and window size adaptation.

REFERENCES

AL-KATEB, M., LEE, B. S., AND WANG, X. S. 2007. Adaptive-Size reservoir sampling over data streams. In *Proceedings of the 19th International Conference on Scientific and Statistical Database Management*. IEEE Computer Society, 22–34.

ARASU, A., BABCOCK, B., BABU, S., DATAR, M., ITO, K., NISHIZAWA, I., ROSENSTEIN, J., AND WIDOM, J. 2003. The stream group. STREAM: The Stanford Stream Data Manager. http://infolab.stanford.edu/stream.

BALLOU, D. P. AND TAYI, G. K. 1999. Enhancing data quality in data warehouse environments. *Comm. ACM 42*, 1, 73–78.

Final edited form was published in "Journal of Data and Information Quality". 2009, 1(2), S. 1-28. ISSN 1936-1963.
https://doi.org/10.1145/1577840.1577845

BURDICK, D., DESHPANDE, P. M., JAYRAM, T. S., RAMAKRISHNAN, R., AND VAITHYANATHAN, S. 2005. Olap over uncertain and imprecise data. In *Proceedings of the 31st International Conference on Very Large Data Bases*. VLDB Endowment, 970–981.

HAAS, P. J. 1997. Large-Sample and deterministic confidence intervals for online aggregation. In *Proceedings of the 9th International Conference on Scientific and Statistical Database Management*. 51–63.

KANG, J., NAUGHTON, J. F., AND VIGLAS, S. D. 2002. Evaluating window joins over unbounded streams. In *Proceedings of the 28th International Conference on Very Large Data Bases*. 341–352.

KLEIN, A. 2007. Incorporating quality aspects in sensor data streams. In *Proceedings of the ACM 1st Ph.D. Workshop in CIKM (PIKM)*. ACM, New York, 77–84.

KLEIN, A., DO, H.-H., AND LEHNER, W. 2007. Representing data quality for streaming and static data. In *Proceedings of the International Workshop on Ambient Intelligence, Media, and Sensing (AIMS)*. AIMS Workshop, 3–10.

KRAEMER, J. AND SEEGER, B. 2004. Pipes - A public infrastructure for processing and exploring streams. In *Proceedings of the 9th International Conference on Management of Data*, G. Weikum et al., eds. ACM, 925–926.

LEE, M.-L., LING, T. W., LU, H., AND KO, Y. T. 1999. Cleansing data for mining and warehousing. In *Proceedings of the 10th International Workshop on Database and Expert Systems Applications*. 751–760.

MIELKE, M., MUELLER, H., AND NAUMANN, F. 2005. Ein data-quality-wettbewerb. *Datenbank-Spektrum 14*, 34–37.

MOON, Y.-S. 2006. Efficient stream sequence matching algorithms for handheld devices on time-series stream data. In *Proceedings of the 24th IASTED International Conference on Database and Applications (DBA)*. ACTA Press, Anaheim, CA, 44–49.

MOTRO, A. AND RAKOV, I. 1996. Estimating the quality of data in relational databases. In *Proceedings of the International Conference on Information Quality (IQ)*. 94–106.

MOTRO, A. AND RAKOV, I. 1997. Not all answers are equally good: Estimating the quality of database answers. In *Flexible Query Answering Systems*. Kluwer Academic Publishers, 1–21.

NAUMANN, F. AND ROLKER, C. 1999. Do metadata models meet IQ requirements? In *Proceedings of the International Conference on Information Quality (IQ)*. 99–114.

NAUMANN, F. AND ROLKER, C. 2000. Assessment methods for information quality criteria. In *Proceedings of the International Conference on Information Quality (IQ)*. 148–162.

ORR, K. 1998. Data quality and systems theory. *Comm. ACM 41*, 2, 66–71.

PAPULA, L. 2006. *Mathematische Formelsammlung fuer Ingenieure und Naturwissen-schaftler* (German). Vieweg Verlag.

SCANNAPIECO, M. AND BATINI, C. 2004. Completeness in the relational model: A comprehensive framework. In *Proceedings of the 9th International Conference on Information Quality (IQ)*. 333–345.

SCHMIDT, S., FIEDLER, M., AND LEHNER, W. 2005. Source-Aware join strategies of sensor data streams. In *Proceedings of the 17th International Conference on Scientific and Statistical Database Management (SSDBM)*. California, 123–132.

STRONG, D. M., LEE, Y. W., AND WANG, R. Y. 1997. Data quality in context. *Comm. ACM 40*, 5, 103–110.

WAND, Y. AND WANG, R. Y. 1996. Anchoring data quality dimensions in ontological foundations. *Comm. ACM 39*, 11, 86–95.

WANG, R. Y. AND STRONG, D. M. 1996. Beyond accuracy: What data quality means to data consumers. *J. Manag. Inform. Syst. 12*, 4, 5–33.

WANG, R., ZIAD, W., AND LEE, Y. 2001. *Data Quality*. The Kluwer International Series on Advances in Database Systems, Vol. 23. 63–77.

WEIKUM, G. 1999. Towards guaranteed quality and dependability of information service. In *Proceedings of the 8th GI Fachtagung: Datenbanksysteme in Buero, Technik und Wissenschaft*, A. P. Buchmann, ed., Springer Verlag, 379–409.