Data Quality Observation in Pervasive Environments

Fei Li, Stefan Nastic, Schahram Dustdar
Distributed Systems Group
Vienna University of Technology
Argentinierstrasse 8/184-1, 1040, Vienna, Austria
Email: {lastname}@infosys.tuwien.ac.at

***Abstract*—Pervasive applications are based on acquisition and consumption of real-time data from various environments. The quality of such data fluctuates constantly because of the dynamic nature of pervasive environments. Although data quality has notable impact on applications, little has been done on handling data quality in such environments. On the one hand past data quality research is mostly in the scope of database applications. On the other hand the work on Quality of Context still lacks feasibility in practice, thus has not yet been adopted by most context-aware systems. This paper proposes three metric definitions—*Currency*, *Availability* and *Validity*—for pervasive applications to quantitatively observe the quality of real-time data and data sources. Compared to previous work, the definitions ensure that all the parameters are interpretable and obtainable. Furthermore, the paper demonstrates the feasibility of proposed metrics by applying them to real-world data sources on open IoT platform Cosm[1] (formerly Pachube).**

***Keywords*-pervasive computing, data quality, real-time data, internet of things**

I. Introduction

Data quality[1][2] problems, for example accuracy, currency, completeness and so on, are inherent to information systems. Past research on data quality has mostly been conducted in the context of database applications, in which data are relatively stable and data schemata are usually well-defined. However, little has been done on quality of real-time data. The emergence of pervasive systems [3], which heavily rely on acquisition and consumption of real-time data, is challenging the traditional methodologies for defining, measuring and improving data quality.

In pervasive systems, data sources are situated in dynamic environments and data are consumed ad hoc by applications. Data quality problems can occur because of heterogeneity of data sources, weak data schemata, hardware or network failures, environmental interferences, etc. Furthermore, the problems are exacerbated by the growing scale and openness of pervasive systems, such as smart cities, opportunistic sensing and social-aware applications, because of the voluntarism and geographical distribution of data sources.

Observing data quality is an essential step towards handling data quality problems and preparing data for further uses. To date, the best known work on data quality in pervasive environments is Quality of Context (QoC)[4], which concerns with the quality of information in context-aware systems. Although various quality definitions have been discussed in the literature, they have not been widely applied to context-aware systems[5]. One fundamental drawback of this research is the lack of feasible metric definitions that can help users to measure context quality in practice. Since data quality reflects both objective measurements on data and their utility to applications, to quantify data quality, the parameters in metrics definitions should be customized for each usage scenario. Therefore, the methods to acquire the parameters ought to be provided along with the quality metric definitions.

To this end, this paper is focused on two essential activities related to data quality in pervasive environments: defining metrics and observing them on real-world data. We propose a *currency* definition that incorporates the dynamic update behavior of each data source. *Availability* of data sources is redefined from users perspective to represent their utility to applications. And *Validity* is defined as a pragmatic approach to verify the attribute values of real-time data. The definitions of these metrics ensure that their parameters are interpretable and obtainable by analyzing historical data. Their feasibility along with corresponding parameterization methods is demonstrated by applying them to observe the quality of two data feeds on open IoT platform Cosm. To the best of our knowledge, this is the first work to apply real-time data quality metrics on open data source in the real world.

The paper is organized as follows: Section II provides a comprehensive analysis of ongoing work with regard to data quality in both database and quality of context. Section III porposes the observable data quality metrics. Then we apply the metrics on real-world data sets and present the results in Section IV. Finally, the paper concludes in Section V.

II. Related work

Data quality in database systems has been investigated for decades. General research framework[11] is now established, and the benefits of the research have been shown in various data intensive activities[2]. The dimensions of data quality have first been systematically studied in the context of management information systems[1], and most of the metrics used later are derived from this research. As one of the most important and fundamental metrics, currency definition has been investigated by many researchers. Let T^{exp} be the valid lifetime of a data object and $Age = t^{cur} - t^{update}$, where t^{cur} and t^{update} are respectively current and update time, Ballou et al.[12] defined $Currency$=$\{\max[0, (1 - \frac{Age}{T^{exp}})]\}^s$. With a slight variation, Even and Shankaranarayanan [13] defined

[1]http://cosm.com. Formerly http://pachube.com

TABLE I
QUALITY OF CONTEXT METRIC COVERAGE

	Precision	Prob. Correctness	Trust-worthiness	Resolution	Up-to-dateness	Completeness	Consistency
Buchholz et al.[4]	*	*	*	*	*		
Bu et al.[6]		*			*		*
Kim et al.[7]	*					*	*
Scheikh et al.[8]	*			*	*		
Manzoor et al.[9]	*		*		*	*	
Neisse et al. [10]	*		*	*	*		

$Currency$= $\max\{0, [1 - (\frac{Age}{T^{exp}})^s]\}$. In both definitions s is supposed to be customized for different applications. However, setting the value of s is arbitrary and the appropriateness of a setting is hard to be verified. Heinrich et al.[14] proposed $Currency$=$e^{-decline(A)*age(w,A)}$, where A is the attribute and w is the current attribute value. Setting a decline function in this definition is still tricky, and it is hard to incorporate expiration time of data. Heinrich et al.[15] compared different definitions of currency in the literature and pointed out six important requirements for data quality metrics to be adopted by applications—Normalization, Interval Scale, Interpretability, Aggregation, Adaptivity, and Feasibility. These requirements are used to guide the development of a currency metric for a campaign management application. Web data have higher change rate, and due to their unstructured nature, it is more difficult to model and measure their quality. Pernici and Scannapieco[16] have focused on the temporal dynamics of web data, and proposed *Volatility* and *Completabiltiy* definitions. However, they are measured based on subjective user feedbacks. This paper deals with the real-time data in pervasive environments. Such type of data is fundamentally different to database applications or web data in two ways. First, real-time data change and expire constantly, because they represent the dynamic attributes in the real world. Second, the ground truth—the actual value of an attribute in the real world—is almost impossible to capture. Because of these two basic characters, the data quality metric definitions proposed in aforementioned research are not applicable in pervasive environments.

On the other hand, in QoC research many metrics have been defined, but their feasibility and utility are questionable. The literature coverage of the most common metrics of QoC is summarized in Table I. The definitions on the first five metrics were originally proposed by Buchholz et al[4] and have generally remained the same in the literature, namely *Precision*, *Probability of Correctness*, *Trust-worthiness*, *Resolution*, *Up-to-dateness*. Among these metrics, Precision and Resolution are objective and observable through well-established measurement methods. However, because the real value of an observed object is not obtainable in pervasive environments, probability of correctness is not feasible in practice. The same applies to accuracy, which is one of the most common metrics in database applications. Trust-worthiness is intrinsically different to the other metrics. It requires feedbacks from data consumers. For a detailed analysis of these four metrics we recommend readers referring to Neisse et al. [10]. *Up-to-dateness* is formally defined by Manzoor et al. [9]: *Up-to-dateness*= $\max[0, (1 - \frac{Age}{T^{exp}})]$. This definition does not adapt to the character of different update behavior of data sources because it is essentially a linear decline function simplified from previously discussed currency definitions.

Completeness is defined by Kim et al.[7] and Manzoor et al.[9] as the ratio of the number of attributes available to the total number of attributes. Manzoor et al.'s completeness definition requires setting weight for each attribute involved, which is highly impractical. Bu et al[6] have defined *inconsistency* as the difference between raw input values from different sources on the same context attribute, and proposed an inconsistency resolution algorithm based on comparing the appearance frequency of each value. Kim et al.[7] have only defined *representation consistency*, which refers to the extent to which data is presented in the same format. Completeness and Consistency are aggregative metrics that are applied when multiple data objects are aggregated in different ways. Since we focus on the feasibility and utility of basic metrics, aggregative metrics are not in the scope of this paper.

Although the QoC research has established some conceptual basis to understand the data quality in pervasive environments, the metric definitions in literature are still facing challenges in applying them to real-world problems. The proposed metric frameworks either provide overly simplistic definitions, or eluded the discussions about necessary parameterization. In addition, among all the discussions on QoC metrics, only Kim et al.[7] and Bu et al.[6] have provided small scale setup or simulation to apply some metrics. In contrast, our work is not intended to propose a general metric framework, as the choice of quality dimensions to be applied should be left to pervasive applications. Alternatively we focus on redefining several metrics that are most commonly used by pervasive applications in practice. Rather than assuming the input parameters are in place for calculation, we propose systematic methods to parameterize the metrics in order to reflect the character of the sources and the requirements of applications. Furthermore, we apply our proposed metrics to real-world data collected from an open pervasive computing platform to demonstrate their feasibility.

III. METRICS

This section presents the definitions of *Currency*, *Availability* and *Validity* for pervasive environments. In order to easily use the metrics in practice, we pay particular attention to the observability of these metrics—the parameters for calculating

the metrics can be obtained by observing the history of a data source, and require minimal user inputs.

A. Currency

Currency represents the temporal utility of a data object after it is created. Intuitively, the utility decreases monolithically until the object is considered not reliably representing reality anymore, i.e., expires. Thus we adopt the linear decline function $\max[0, (1-\frac{Age}{T^{exp}})]$ as the basis for currency definition because it is generally applicable and normalized to $[0, 1]$. As in most work on temporal attributes of data, T^{exp} shall be provided by domain experts according to the data type and its application scenario. We consider providing this parameter a reasonable requirement to data consumers.

The utility of an object declines when there is a newer object available to represent the same attribute in the real world. Therefore understanding the dynamic behavior of data updates is also important to the definition of currency. To this end, we first introduce an update function—$f^{update}(t)$. Let T be the interval between two consecutive data updates from a certain data source. $T \in (0, +\infty)$ is a random variable, of which we define $f^{update}(t)$ as the probability density function. Thus $P[a \leq T \leq b] = \int_a^b f^{update}(t)\,\mathrm{d}t$ is the probability that a data update happens between time a and b. In practice, f^{update} can be observed by building histogram from historical data. We will demonstrate the approach in the next section.

Based on the update function, we introduce *Volatility* of data objects. Volatility is the probability of an update to happen between the last update (time point 0) and the current time (Age of the current object), defined in (1).

$$Volatility = \int_0^{Age} f^{update}(t)\,\mathrm{d}t \tag{1}$$

Using volatility as a scaling factor for currency, we propose a currency definition for pervasive applications in (2).

$$Currency = (1 - \frac{Age}{T^{exp}}) * e^{-Volatility}, T^{exp} \geq Age \tag{2}$$

When $Volatility = 1$, which means a new data object is definitely available, $Currency = (1 - \frac{Age}{T^{exp}})/e$. In this case, the utility of data is reduced because data consumer should be able to obtain the newer object. When $Volatility = 0$, i.e. there is certainly no update so far, the object currency decreases linearly with regard to its age. More generally, when volatility is high, currency is scaled down because it is likely that the data consumer can get an update. It is also worth noting that when necessary, the linear decline function can be replaced by other functions that suit a specific application scenario, but the scaling effect of volatility is still applicable and preserved.

The advantage of the proposed currency definition can be evaluated according to the six requirements proposed by Heinrich et al.[15] for data quality attributes.

1) *Normalization* The metric is normalized to $[0, 1]$. When $Age = 0$, $Currency = 1$; when $Age = T^{exp}$, $Currency = 0$. The definition also complies to general principle that when the attribute will not change at all, for example date of birth, for which $Volatility = 0$ and $T^{exp} \to \infty$, the currency is always 1.
2) *Interval Scale* For the same f^{update} function, the same difference between two levels of currency means the same extent of improvement. For example, a difference of 0.1 between 0.2 and 0.3 and between 0.5 and 0.6 can be understood as the same extent of utility improvement.
3) *Interpretability* The metric can be easily interpreted because the input parameters are interpretable. The metric avoids setting obscure empirical parameter to scale currency in each specific usage scenario.
4) *Aggregation* In pervasive environments, real-time data are not aggregated into tuples or relations, but into new events according to usage scenario of each specific application. Correspondingly, the currency of the aggregated events can be defined—for example, by the newest event, by the oldest event, or by the average time—depending on the application requirements. This paper is not focused on metric aggregation, but we will investigate this direction in the future work.
5) *Adaptivity* The metric can be adapted to each specific source and application by customizing the update function and expiration time.
6) *Feasibility* The input parameters are determinable because the f^{update} can be statistically determined and T^{exp} can be set by applications according to user requirements.

B. Availability

Availability is a typical quality metric for data sources. From utility's perspective, it is indifferent to an application if a data source is online or not as long as there is an unexpired data object from the data source. Conversely, if a data source is online but unable to provide the data that are current enough for an application, it is effectively the same as unavailable. Therefore, we define availability for real-time data as follows: given an observation period OP, the availability of a data source is the percentage of the time that there is an unexpired data object provided by the source. Formally, let n be the total number of objects received in OP, the availability is defined in (3), where t_i is the interval between the ith and the $i+1$th updates (t_n can be defined as the time between the end of OP and the update time of the nth object).

$$Availability = 1 - \frac{\sum_{i=i}^{n} \max(0, t_i - T^{exp})}{OP} \tag{3}$$

When an interval t_i exceeds the expiration time T^{exp}, the unavailable time $t_i - T^{exp}$ will be added. Otherwise the unavailable time before next update is 0. Compared to the usual availability definition based on the online time of data sources, (3) is easily observable as the availability of data source is perceived by applications through the arrival of data objects.

C. Validity of data

In database applications, the correct value of an attribute, e.g. home address, birthday, employment status, can usually be verified in the real world or validated by cross-checking multiple data sources using object identification techniques[2]. However, pervasive environments are fluid, and data objects are transient, e.g., temperature, location. In most cases it is impossible to identify the baseline (real-world value of an attribute at the moment of observation) to judge the correctness or accuracy of data.

Therefore, we take a pragmatic approach to defining the correctness of data. The attribute observed by a source is considered correct as long as we can estimate, with certain level of confidence, that the observation does not deviate from the real-world situation beyond an acceptable range. We refer to this metric as validity.

Validity is a set of constraints to the data used in a certain application scenario. It consists of the properties, expressed as rules that need to be satisfied by all the data objects from a data source. A validity rule can be presented as a boolean function: $VR(o) = true$, if rule VR is satisfied by data object o. Otherwise $VR(o) = false$. Validity is defined in (4), where m is the number of rules.

$$Validity = \bigwedge_{i=1}^{m} VR_i(o) \tag{4}$$

The actual rules are decided by the domain-specific properties of a certain attribute and its application scenario. In general, two types of rules can be applied:

- *Static rules* that can be validated by checking a single data instance. For example, the environmental temperature of Vienna in May is between $0°$C and $35°$C.
- *Dynamic rules* that are mainly used to validate if the changes of data are reasonable. For example, a drop of 10 degrees in environmental temperature cannot happen in half an hour.

Validity can be extended to evaluate the historical performance of data sources. Let n be the total number of updates in observation period OP, and n^{valid} be the valid instances updated in P, the probability of validity is defined in (5).

$$Prob.Validity = \frac{n^{valid}}{n} \tag{5}$$

IV. Quality observation

In this section we demonstrate how to apply the metrics on observing data quality of real-world data available on Cosm platform.

Cosm is an open IoT platform on which sensor owners world-wide can provide their live sensory data streams (feeds) and open them on the web to third party application developers. Since the provisioning of data is completely voluntary, the feeds are provided without commitment, and there is no explicit quality assurance mechanism. In fact, randomly browsing through the feeds, one can easily notice that a considerable amount of feeds are not alive, or their data are not up to date. In addition, although the total number of feeds is large, the feeds are still geographically sparse. It is very hard to find two feeds updating one same or nearby real-world attribute so that the two feeds could be used for cross-checking. Therefore, the open IoT platform poses significant challenges to application developers on using the feeds. This is also evident from the fact that the growth of applications[2] on Cosm is left far behind the growth of feeds.

Among the live feeds, we chose two relatively stable and meaningful ones to apply our quality metrics. The feeds and observation periods are illustrated in Table II. Feeds can be accessed at http://cosm.com/feeds/[Env ID].

A. Average speed of ship Lena

The data set 1 is a mobile environment located on ship Lena. It is equipped with several data sources reporting Lenas current location, average speed, destination, etc. We take feed 2, average speed of the ship, for analysis. Data were gathered on 14th and 15th May 2012, during which a total of 576 data objects are received. Figure 1 visualizes the speed readings output by this data source during the observation period, and Figure 2 illustrates the update intervals. For the clarity of illustration, both Figure 1 and 2 are plotted with every 10th object.

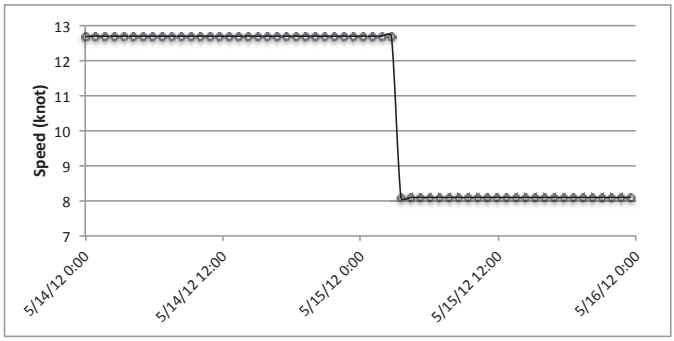


Fig. 1. Feed 3824.2, average speed of ship Lena

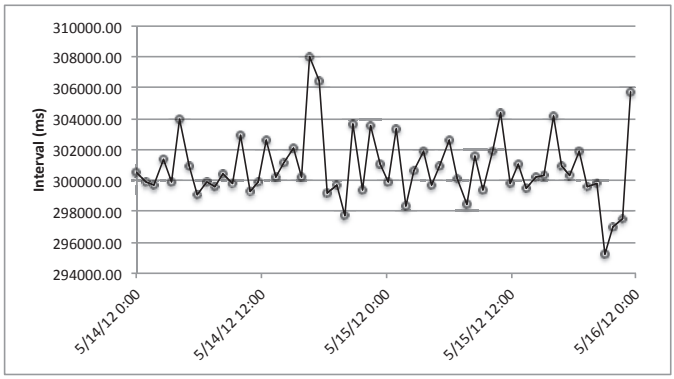


Fig. 2. Feed 3824.2, update interval

[2]http://cosm.com/apps/

TABLE II
COSM DATA SETS USED IN EXPERIMENT

Dataset ID	Env ID	Feed ID, Tag	Description	Location	Exposure	Disposition
1	3824	2, Average Speed	Speed of Ship Lena	Oceans	Outdoor	Mobile
2	504	1, Light level	Pachube office environment	London	Indoor	Fixed

The speed readings are very simple. There are only two values (12.7 and 8.1) and one speed change, which are not abnormal according to common sense. Thus for the speed of ship Lena, we will not analyze the validity of data because of its obvious normality. We will focus on its currency and availability.

1) Currency: From the update interval data we get $min = 288471.00$, $max = 311357.00$, mean $\overline{x} = 300000.25$ and standard deviation $\hat{\sigma} = 2646.80$. First we try to find out f^{update}. We assume that the update interval of this feed follows normal distribution (null hypothesis, H_0). Thus we use Scott's normal reference rule [17] to decide the bin size for building the histogram: $h = \frac{3.5\hat{\sigma}}{n^{1/3}}$, where n is the total number of samples. The result histogram, illustrated in Figure 3, visually supports our assumption. Then we use normal probability plot that calculates the coefficient of determination ($\mathcal{R}^2$) between rank-based z-score of standard normal distribution and standardized ($Z = \frac{X-\overline{x}}{\hat{\sigma}}$) interval data (The plot itself is not shown due to space limit). The result is $R^2 = 0.945$, indicating that we can accept the H_0 that the update interval follows a normal distribution where $\mu = 300000.25$ and $\sigma = 2646.80$. Figure 3 also illustrates its volatility. For this feed, it is the value of the cumulative distribution function.

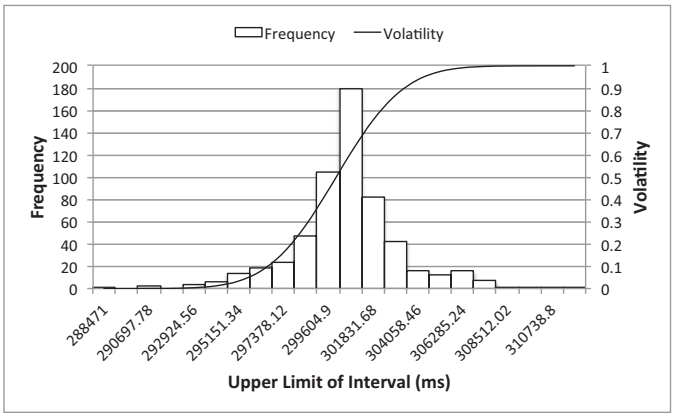


Fig. 3. Feed 3824.2, histogram and volatility

Figure 4 depicts the changes of currency with regard to the ages of objects. The effects of different factors on the currency of an object are clearly illustrated. Expiration time T^{exp} decides in general how fast the currency declines and when it becomes 0. The changes in volatility, or the probability that a newer object is available, are reflected by the non-linear drops of currency during the period when update is most likely to happen.

At last, we apply the currency evaluation approach to the last 20 updates of the observation period, illustrated in Figure 5. The horizontal line on the figure indicates $T^{exp} = 305000$. The observed update behavior can be applied continuously to new data objects. Conversely, new data objects can also be used to continuously tune the update function.

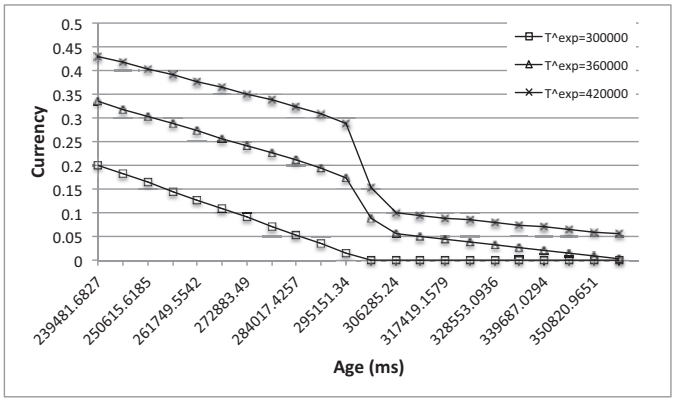


Fig. 4. Feed 3824.2, currency changes with different T^{exp}

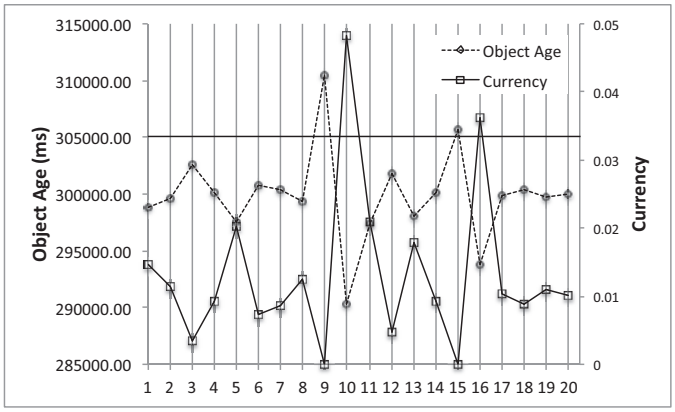


Fig. 5. Feed 3824.2, object currency

2) Availability: According to our definition in (3) availability of a data source is subject to the expiration time of data objects. In other words, the perception of availability is relative and dependent on the expectation of data users. Table III demonstrates the availability of this data source in the observation period with regard to different T^{exp} settings.

TABLE III
AVAILABILITY OF FEED 3824.2

$T^{exp}(ms)$	Availability
240000	79.965%
290000	96.660%
300000	99.701%
310000	99.999%

B. Light level of Pachube office

The environment 504 is fixed in Pachube office. A set of basic environmental attributes are measured, including temperature, humidity and so on. We take feed 2, light level of the office, for analysis. The data were gathered from 19th to 25th February 2012, during which a total of 9094 data objects are updated. Figure 6 illustrates the light level readings during the observation time, and Figure 7 illustrates that the update intervals. Every 50th object is plotted.

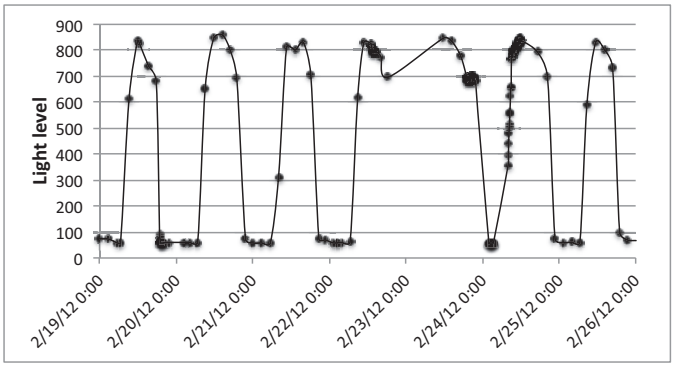


Fig. 6. Feed 504.1, light level

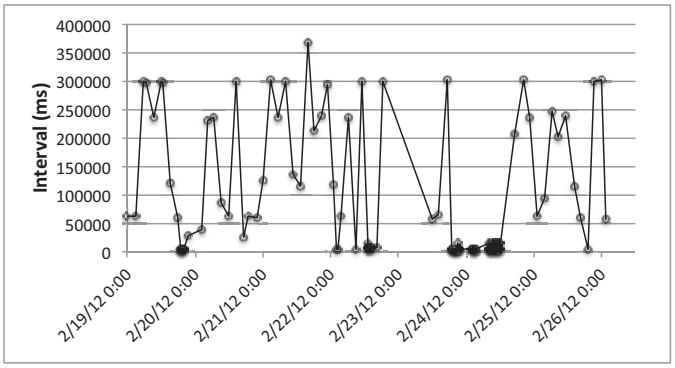
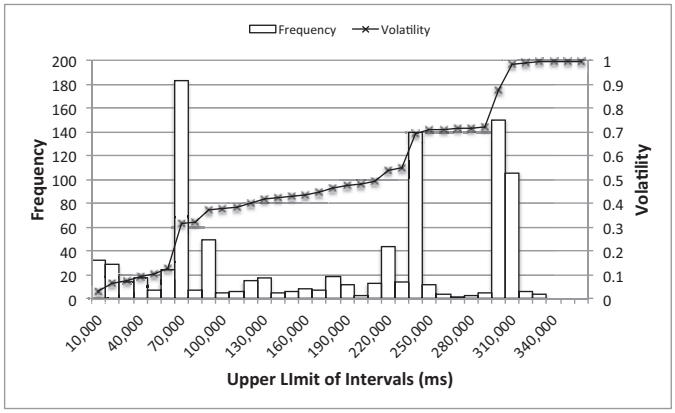


Fig. 7. Feed 504.1, update interval

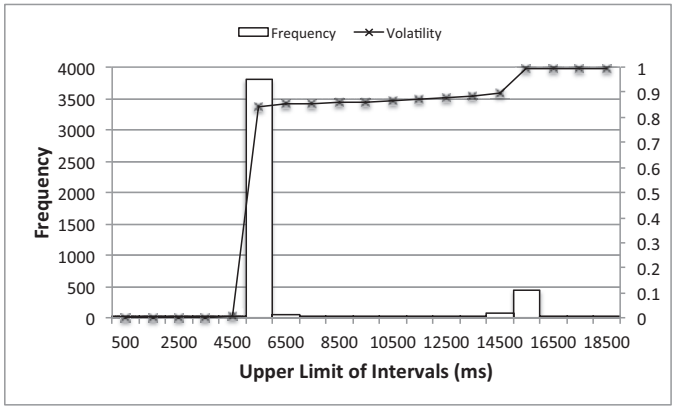
From Figure 6 we can observe that the light level generally corresponds to day and night. Also worth noting is that there are multiple random periods of time the updates are missing. They will have impact to the analysis of availability. Update intervals are distributed in a very wide range. From Figure 7, the update intervals between 50 seconds and 300 seconds can easily be seen. But there are also several periods where the data are mostly updated at a 5 seconds interval. They are the concentrated data points at the bottom of Figure 7.

From the original data, we can intuitively identify two distinct update behaviors that switch between each other randomly. Without any knowledge of the data source configuration and its working environment, it is not possible to identify the reasons or find out any regularity in this behavior. We can only assume that two sensor configurations are in the environment. Thus we take two separate observation periods (OP) from the whole data set: OP1=[Feb. 20 2012 0:01, Feb. 21 2012 23:56] with 978 objects and OP2=[Feb. 23 2012 19:02, Feb. 24 2012 11:59] with 4541 objects. Each of them represents one update behavior, of which the mean value $\overline{x}^{OP1} = 176446.65$, $\overline{x}^{OP2} = 9867.96$. In practice, the switch between different update behaviors can be detected by monitoring the interval mean of sliding window. With this rather irregular feed we intend to demonstrate: to quantitatively analyze the quality of data for the benefits of users, it is unnecessary to acquire knowledge about the data source settings. This is a very important requirement in employing data from open environments.

1) Currency: The histograms of the samples in two OPs are illustrated in Figure 8.



(a) OP1



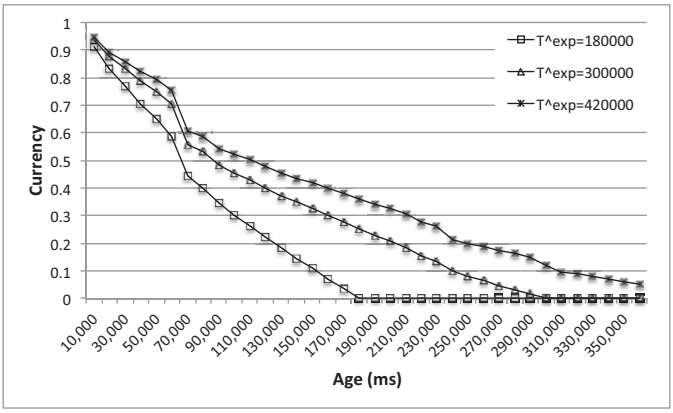
(b) OP2

Fig. 8. Feed 504.1, histogram and volatility

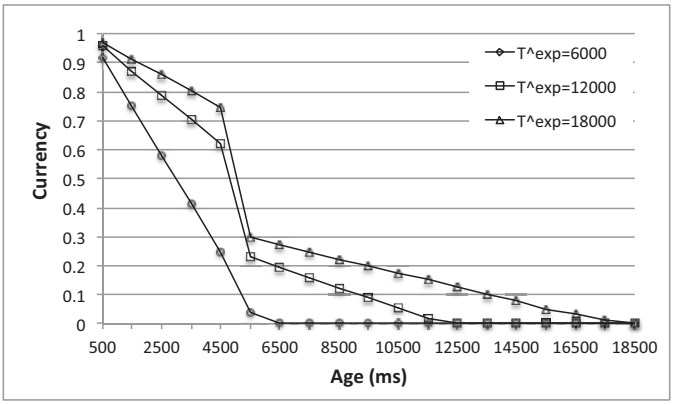
In OP1, most intervals are between 10 seconds and 310 seconds, with several intervals more frequently appearing—60, 240 and 300 seconds. However, none of these more likely intervals appear in a significant number of consecutive updates. In fact, the intervals switch between each other randomly. Therefore, we can not draw a conclusion on any distribution that might taken these intervals as mean value. In OP2, most updates happen at a 5-second interval. But a notable amount of updates also happen at an interval of 15 seconds. In other words, when missing objects, two are usually missing continuously. The change of volatility with regard to the

update intervals, particularly those high frequency intervals, can also be seen on Figure 8. Since formal update functions are not available, the volatility is calculated based on relative frequency of intervals in the respective observation period.

The changes of data currency are illustrated in Figure 9. For demonstration, we set the expiration time differently for the two OPs. Scaling effect of volatility on currency can easily be spotted. Due to space limitation, we do not illustrate the observation on individual objects from this data source. The approach is clearly demonstrated by dataset 1.



(a) OP1



(b) OP2

Fig. 9. Feed 504.1, currency

2) Availability: Table IV displays the availability of two observation periods with regard to different T^{exp} settings. In OP2, the availability is relatively low because there are no updates between Feb. 24 02:00:04 and Feb. 24 08:00:00.

TABLE IV
FEED 504.1, AVAILABILITY

OP1		OP2	
$T^{exp}(ms)$	Availability	$T^{exp}(ms)$	Availability
120000	39.98%	6000	64.85%
240000	69.53%	12000	71.01%
360000	99.39%	18000	73.64%

3) Validity: Figure 6 has demonstrated that the light levels are high during work hours and changes are in a wider range. The changes can be triggered by many reasons, e.g. turning lights off and on and shifts of daylight. Conversely during the night light levels are low and changes are small. Thus we assume two basic rules for the validity of light level. First is that normal light level should be in a certain range. In fact, Cosm has provided the API for data provider to filter data according to min and max threshold. The data we collected on Cosm platform should already satisfy this rule. The second rule is that the changes of light level between two objects should be in a certain range. Because of the intensity of human behavior are different during day and night, we set rules separately it for day [7:30-19:30] and night [19:30-7:30]. The threshold for all rules is set to $\overline{x} + 3\sigma$. It is worth noting that in this experiment we only intend to demonstrate the use of validity rules. Better models of rules and parameterization should be based on how an application models the environment.

We observe the volatility in the period of [Feb. 25 2012 12:00:00, Feb. 26 2012 3:56:00], during which a total of 571 objects are updated. The observation is summarized in Table V. The mean and standard deviation of data are calculated based on the whole data set 2. There are 17 objects violated light change rule of day. Then the probability of validity in the observation period is 97.02%.

TABLE V
VALIDITY OF FEED 504.1

Rules	Condition	$\overline{x}$	σ	Threshold	Violated
Light Range	All time	499.68	331.26	1493.45	0
Light Change	Day	4.72	13.16	44.20	17
Light Change	Night	2.35	3.83	13.84	0

C. Remarks on the observational experiments

The update behaviors of data sources in pervasive environments are very diverse. Even for one data source, its behavior can change randomly. To understand the behavior of a source requires certain knowledge of the source's configurations and its working environment. However, for applications that intend to utilize an open data source, the most important is not to find the reasons behind various behaviors, but to quantitatively measure the utility of the data source and its produced data objects. It is possible that f^{update} is not always of certain distributions that can be parameterized through statistical methods, as in the case of data set 2, but to present the function itself is not required in practice. The proposed observation methods are feasible as long as there is a sufficient amount of historical data because we can directly calculate the relative frequencies to obtain volatility.

To set rules for validity is more subjective, and the understanding to the interested attribute is very important for setting the right rules. Our purpose of this paper is to argue against the common misconception of including correctness and accuracy as quality metrics for pervasive environments, and to propose a notion of validity that is feasible in practice. Therefore we exemplify several typical validity rules and possible ways to parameterize them, but these rules are not to be regarded as

generic or exhaustive. The validity rules are always application and attribute-specific. There is no one-size-fits-all solution in data quality[18]. Other than the data attribute itself, contextual information can also be useful for setting the correct rules. In our example, if the presence information of people in the office were available, we could have set more appropriate conditions for the validity of light levels.

Last but not least, interpretation is important in many data analysis work. Data quality observation is no exception. Taking validity as an example again, when a data object exceeds the preset range, it is eventually up to users to decide how to use the low quality data object.

V. Conclusion

The paper defined metrics for data quality in pervasive environments and applied them on real-world data sources to demonstrate the feasibility of the metrics. In previous research, the data quality work in database applications was not applicable to pervasive environments, while the metrics proposed in QoC research were either unobservable or unadaptable to application requirements. Therefore, we redefined three metrics for pervasive environments, namely Currency, Availability and Validity, in a way that all parameters are observable and easily understood by data consumers. We demonstrated the feasibility of the metrics by applying them to two Cosm data sources. One is a relatively stable data source with regular readings and normally distributed update intervals, whereas another is heavily interfered by human activity and other unknown configuration conditions. At last we discussed the experiences gained from the observational experiments.

Our future work will be carried out in three directions. First is to develop aggregative quality metrics that can be applied in more complex data processing schemes, such as data composition and selection. Second is to automate the data quality observation process in a large-scale pervasive application platform based on our previous study on context provisioning[19]. The third and the eventual goal of this research is to develop a set of mechanisms to assuring and improving data quality for pervasive applications.

Acknowledgment

This work is sponsored by Pacific Controls Cloud Computing Lab (PC^3L), a joint lab between Pacific Controls L.L.C., Scheikh Zayed Road, Dubai, United Arab Emirates and the Distributed Systems Group of the Vienna University of Technology.

References

[1] R. Y. Wang and D. M. Strong, "Beyond accuracy: what data quality means to data consumers," *Journal of Management Information Systems*, vol. 12, no. 4, pp. 5–33, Mar. 1996. [Online]. Available: http://dl.acm.org/citation.cfm?id=1189570.1189572

[2] C. Batini and M. Scannapieco, *Data Quality-Concepts, Methodologies and Techniques.* Springer Berlin Heidelberg New York, 2006.

[3] M. Satyanarayanan, "Pervasive computing: vision and challenges," *IEEE Personal Communications*, vol. 8, no. 4, pp. 10–17, 2001. [Online]. Available: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=943998

[4] T. Buchholz, A. Kupper, and M. Schiffers, "Quality of context: What it is and why we need it," 2003.

[5] P. Bellavista, C. Antonio, F. MARIO, and L. FOSCHINI, "A Survey of Context Data Distribution for Mobile Ubiquitous Systems," *ACM Computing Surveys*, vol. 45, no. 1, 2013. [Online]. Available: http://www-lia.deis.unibo.it/Staff/LucaFoschini/pdfDocs/context_survey_CSUR.pdf

[6] Y. Bu, T. Gu, X. Tao, J. Li, S. Chen, and J. Lu, "Managing Quality of Context in Pervasive Computing," in *2006 Sixth International Conference on Quality Software (QSIC'06)*. IEEE, Oct. 2006, pp. 193–200. [Online]. Available: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4032285

[7] Y. Kim and K. Lee, "A Quality Measurement Method of Context Information in Ubiquitous Environments," in *2006 International Conference on Hybrid Information Technology*. IEEE, Nov. 2006, pp. 576–581. [Online]. Available: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4021269

[8] K. Sheikh, M. Wegdam, and M. van Sinderen, "Middleware Support for Quality of Context in Pervasive Context-Aware Systems," in *Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerComW'07)*. IEEE, Mar. 2007, pp. 461–466. [Online]. Available: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4144879

[9] A. Manzoor, H. L. Truong, and S. Dustdar, "On the Evaluation of Quality of Context," 2008, pp. 140–153.

[10] R. Neisse, M. Wegdam, and M. van Sinderen, "Trustworthiness and Quality of Context Information," in *2008 The 9th International Conference for Young Computer Scientists*. IEEE, Nov. 2008, pp. 1925–1931. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4709268

[11] S. E. Madnick, R. Y. Wang, Y. W. Lee, and H. Zhu, "Overview and Framework for Data and Information Quality Research," *Journal of Data and Information Quality*, vol. 1, no. 1, pp. 1–22, Jun. 2009. [Online]. Available: http://dl.acm.org/citation.cfm?id=1515693.1516680

[12] D. Ballou, R. Wang, H. Pazer, and G. K. Tayi, "Modeling Information Manufacturing Systems to Determine Information Product Quality," *Management Science*, vol. 44, no. 4, pp. 462–484, Apr. 1998. [Online]. Available: http://mansci.journal.informs.org/cgi/content/abstract/44/4/462

[13] A. Even and G. Shankaranarayanan, "Utility-driven assessment of data quality," *ACM SIGMIS Database*, vol. 38, no. 2, p. 75, May 2007. [Online]. Available: http://dl.acm.org/citation.cfm?id=1240616.1240623

[14] B. Heinrich, M. Kaiser, and M. Klier, "How to measure data quality?-A metric based approach," in *Proceedings of the 28th International Conference on Information Systems*, 2007.

[15] B. Heinrich, M. Klier, and M. Kaiser, "A Procedure to Develop Metrics for Currency and its Application in CRM," *Journal of Data and Information Quality*, vol. 1, no. 1, pp. 1–28, Jun. 2009. [Online]. Available: http://dl.acm.org/citation.cfm?id=1515693.1515697

[16] B. Pernici, M. Scannapieco, S. Spaccapietra, S. March, and K. Aberer, "Data Quality in Web Information Systems," *Journal on Data Semantics I*, vol. 2800, pp. 48–68, 2003. [Online]. Available: http://www.springerlink.com/content/rn1yyj6c04r2k6lx/

[17] D. W. SCOTT, "On optimal and data-based histograms," *Biometrika*, vol. 66, no. 3, pp. 605–610, Dec. 1979. [Online]. Available: http://biomet.oxfordjournals.org/cgi/content/abstract/66/3/605

[18] L. L. Pipino, Y. W. Lee, and R. Y. Wang, "Data quality assessment," *Communications of the ACM*, vol. 45, no. 4, p. 211, Apr. 2002. [Online]. Available: http://dl.acm.org/citation.cfm?id=505248.506010

[19] F. Li, S. Sehic, and S. Dustdar, "COPAL: An adaptive approach to context provisioning," in *2010 IEEE 6th International Conference on Wireless and Mobile Computing, Networking and Communications*. IEEE, Oct. 2010, pp. 286–293. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5645051