Quality of Context: Models and Applications for Context-aware Systems in Pervasive Environments

Atif Manzoor, Hong-Linh Truong, Schahram Dustdar

Vienna University of Technology, Institute of Information Systems, Distributed Systems Group, Argentinier-strasse 8, A-1040 Vienna, Austria.
E-mail: {manzoor, truong, dustdar}@infosys.tuwien.ac.at

Abstract

Limitations of sensors and the situation of a specific measurement can affect the quality of context information that is implicitly collected in pervasive environments. The lack of information about Quality of Context (QoC) can result in degraded performance of context-aware systems in pervasive environments, without knowing the actual problem. Context-aware systems can take advantage of QoC if context producers also provide QoC metrics along with context information. In this paper, we analyze QoC and present our model for processing QoC metrics. We evaluate QoC metrics considering the capabilities of sensors, circumstances of specific measurement, requirements of context consumer, and the situation of the use of context information. We also illustrate how QoC metrics can facilitate in enhancing the effectiveness and efficiency of different tasks performed by a system to provide context information in pervasive environments.

Key-words: Quality of Context, context-aware systems, pervasive environments.

1   Introduction

Pervasive environments are characterized by a plethora of computing and communication enabled devices that diffuse themselves in everyday living and become invisible (Weiser (1991)). These devices implicitly sense and provide context, a core task in making a system adaptable, that is far more complicated than explicit input to the system (Gray et al. (2001), Mostefaoui et al. (2004)). Quality of context information is deteriorated during this process and contrary to general assumption context information can be incomplete, inaccurate, and ambiguous (Dey (2001), Henricksen et al. (2002)). Inadequate quality of context information severely influences the adaptiveness of context-aware applications (Chen et al. (2002), Dey (2001), Henricksen et al. (2004)). Context-aware applications also perform extra effort to cope with uncertainty of context information(Ranganathan et al. (2004)). Quality of Context (QoC), a measurable metric that provides information about the quality of context, can help resolving uncertain and conflicting situations about context information. Therefore, context-aware applications can take advantage of QoC if they are provided with usable QoC metrics that are evaluated considering their requirements regarding the collection, processing, and provision of context information. Currently there is not only a lack of solutions that evaluate QoC metrics and pass them along with context information to context consumers, but also existing definitions of QoC ignore its multi facetted nature and consider it as an objective term.

    In this paper we consider both objective and subjective views of QoC and redefine QoC. The objective view of QoC presents quality of context information independent of the requirements of a context consumer while the subjective view of QoC considers quality of context information as its worth for a specific context consumer. We also present our model for processing QoC metrics. We

utilize sensor characteristics, measurement context, and specifications and user requirements to evaluate QoC metrics. Later we use these QoC metrics to perform context aggregation and source selection in a middleware solution to provide context information in pervasive environments. We also illustrate how QoC metrics can help to improve the performance of above mentioned tasks.

Following are our key contributions in this paper:

- Comprehensive definition of QoC.
- Model for precessing QoC metrics.
- Evaluation of QoC metrics considering objective and subjective view of QoC.
- Middleware that provides QoC metrics along with context information.

We organize our paper as follows: Section 2 illustrates the motivating scenario for this work. Section 3 presents an analysis of the existing QoC definitions and describes our comprehensive definition. Section 4 discusses our layered model for processing QoC information. Section 5 describes the procedure of the evaluation of QoC metrics. Section 6 presents the design and implementation of our middleware solution that provides QoC metrics along with context information. Section 7 presents the evaluation of the performance of different tasks performed considering QoC metrics and a discussion about the results. An overview of the related work is presented in Section 8. Finally we conclude our work and give an account of future work in Section 9.

2   Motivation and Scenario

"In August 2002, widespread persistent rain led to the catastrophic floods in many parts of Central Europe. There were extreme rainfall events in Austria on numerous rivers north of Central Alps starting from the west. The northern Federal Provinces of Upper and Lower Austria as well as the Federal Province of Salzburg were particularly affected. This event brought rainfall of extraordinary extent and flood recurrence intervals from several years to more than 100 years. Loses of human life and livestock, damages of the infrastructure, buildings, public and private properties rose the public awareness and the demand for the improvement of future flood mitigation measures, innovative alert systems and new technological solutions needed to improve the rescue activities and the analysis of the damage caused by floods."

This situation is described in (Formayer (2004)) and is supported in (The EU Project WORKPAD (2007)). This project aim at facilitating people, performing rescue work in such situations, by providing context information to them. Field workers, participating in the rescue activities, are equipped with mobile devices. They also collect and share context among themselves and send it to the back end by using context management services on mobile devices as described in our previous works ESCAPE (Truong et al. (2007)) and COSINE (Juszczyk et al. (2009)). But due to the unawareness about the quality of context people face difficulties in using this information. For example, in the aforementioned situation, a flood data analyst can make an analysis of the damage caused by the flood and provide this information to the organizations participating in the rescue activities. He can receive context from the field workers and combines it with the existing data to update the current flood situation. As there are usually more than one context source providing information of the same entity in the field, he has to make analysis of all data to be aware of the quality of context. Contents of all context objects seem vital and he may not been able to select one of them and relate it to the existing data. These problems may not only cost him time and effort, but also affect the quality of his work. Context enriched with QoC metrics such as reliability, timeliness, completeness, and significance can allow him to know the quality of context without looking at the contents of context information and thus can substantially improves his work.

3 Quality of Context

Quality of context has been considered unsatisfactory since the start of research in context-aware systems. Early context-aware systems have also tried to collect and model additional data with context. Subsequently, more research efforts were undertaken to explore the problems associated with the imperfection of context and term Quality of Context (QoC) was coined. QoC was first defined by Buchholz et al. (Buchholz et al. (2003)) as "QoC is any information that describes the quality of information that is used as context information. Thus, QoC refers to information and not the process not the hardware components that possibly provide the information". Later Kraüse et al. (Kraüse et al. (2005)) has also defined QoC as "QoC is any inherent information that describes context information and can be used to determine the worth of the information for a specific application."

Both of the above mentioned definitions consider QoC as an objective term that is independent of the situation of the use of context information and consumer requirements for that context information. However, general quality literature has taken quality as both absolute and relative term and defined quality as "freedom from errors", "conformance to specifications", and "features of product that meet customer's needs"(Juran et al. (1998)). These definitions describe quality as a twofold concept that has an absolute quality, showing that the end product is free of errors and a relative quality that shows how much the end product meets customer's needs. Similarly QoC should also inform about the quality of context information in both of these aspects. First, QoC considers the limitations of sensors in collection of context information and situation of specific measurement, e.g., measurement errors, collection of partial information. This aspect of QoC shows the absolute quality of context information that depicts that how much context information is free of error and describes the current situation in the environment. Second, QoC also considers the fact that different context consumers may have different requirements about quality of context information. This aspect of QoC shows the relative quality of context information that depicts that how much context information meets the requirements of a specific consumer to use it for a specific purpose.

For example, a context information service that provides information about the location of a person uses a GSM method and can provide location information with the granularity of the current district of a person. Another context information service that uses a GPS method provides the location information of that person with the granularity of the current street of the location of a person. As these two services are collecting context information with high accuracy, they will have same objective quality. Context information provided by these services will also have same subjective or relative quality for a context consumer service that is only providing information to tourists about interesting places to visit in a city. But the first service will have a low subjective quality for a context consumer service that makes an optimal plan for visiting all sites in a city, as this service will need the location information of a person with higher granularity, i.e., at least with the detail of the current street of the location of that person. The second context information service that provides location information with higher granularity will have higher subjective quality for this context consumer. This example shows that the quality of context information may vary with different context consumers and QoC cannot be measured independently of a context consumer and the intended purpose. Context that is appropriate for use with one application may not be suitable for use by another application. Therefore QoC must also consider the requirements of context consumer and intended use of the context information. This aspect of QoC will show that how much context information is suitable for use by a specific context consumer for intended purpose.

Considering this objective and subjective nature of QoC we have defined it as,

"Quality of Context indicates the degree of conformity of the context collected by sensors to the prevailing situation in the environment and the requirements of a particular context consumer".

The objective view of QoC considers those features of context that are independent of any requirements of a context consumer or the situation of the use of context. These objective characteristics portray how much context is free from errors, i.e., degree of conformity of context to the prevailing situation in the environment. Information about the sensor characteristics that have collected context and situation of specific measurement will be used to determine objective QoC metrics. Subjective view of QoC shows the characteristics of context that illustrate how much a piece of context meets the requirements of a particular consumer to use it for a specific purpose, i.e., degree of conformity of context to the requirements of a particular context consumer. Information about the intended use and the consumer requirements will be used to determine the subjective metrics of QoC.

4   QoC Processing Model

The QoC processing model shown in Figure 1 presents different layers for processing QoC information. The lowest layer is the QoC source layer that consists of data used by higher layer to evaluate QoC metrics. This layer consists of the characteristics of sensors that collect context information, i.e., *Sensor Characteristics*, situation of a specific measurement, i.e., *Measurement Context*, and the information about the requirements of a context consumer and the detail of the context of the use of information, i.e., *Specifications and Consumer Requirements*. QoC metrics evaluated from the data at QoC source layer lie at the next higher layer and are divided in *Objective QoC Metrics* and *Subjective QoC Metrics*. *Objective QoC Metrics* show the quality of context as an independent quantity and their calculation will involve *Sensor Characteristics* and *Measurement Context*. *Subjective QoC Metrics* will show the quality of context for use by a specific context consumer for particular purpose. These metrics will also involve *Specifications and Consumer Requirements* for their calculations. In the following section we will give the detail of each building block of our QoC processing model.
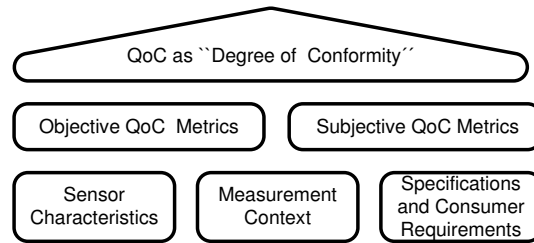
Figure 1   QoC processing model

4.1   Sensor Characteristics

Sensors in pervasive environments are not only limited to physical sensors, such as GPS sensors or temperature sensors, but also include logical or virtual sensors, such as applications that extract the high level context from sensor data and user interfaces to enter information. These sensors may physically be static and fixed at one place or dynamic and move from one place to other place, e.g., sensors mounted on a mobile device. Sensor characteristics will include information about the sensors that can affect the quality of context information provided by those sensors. These characteristics will include the information about the accuracy, granularity, and resolution with which a sensor can collect context. Table 1 presents a brief description of sensor characteristics.

Accuracy is the degree of the correctness of context, i.e., how close is the measured or calculated or collected context information to the actual or true situation in the real world. *Granularity* indicates the degree of detail with which a sensor can collect context information. For example, the location of a person can be expressed at the level of detail of country, city, street, building, floor (Dorn et al. (2006)). Fine granularity or higher value of granularity indicates that information is expressed with more detail. *Precision* is the degree of the exactness of measurement. *Time Period* indicates the time interval between two measurements. For example, if a location sensor collects the location information of a person after every one minute then *Time Period* of this information will be one minute. *Sensor State* indicates whether the source of information is dynamic or static. For example, sensors measuring temperature are fixed at different places in a city. These sources have static value for the *Sensor State*. While sensors embedded in a portable device carried by a human have dynamic value for the *Sensor State*, e.g., GPS sensors embedded in mobile phones. *Sensor Span* is the maximum distance for which a sensor can collect a context object. Every sensor will have a different value of *Sensor Span*. For example, images of a disaster stricken site can be collected by a satellite and an ordinary camera. But the value of *Sensor Span* will be very high for satellite camera as compared to a camera carried by a field worker on disaster site. Similarly, cameras with different capabilities will have different values of sensor span.

Classification	*Metric*	*Definition*
Sensor Characteristics	Accuracy	Extent to which data is correct and free of errors
	Precision	Degree of exactness with which context is collected
	Granularity	Degree of detail with which context is collected
	Time Period	Time interval between two readings of context
	Sensor State	Physical state of sensor
	Sensor Range / Span	Maximum distance for which sensor can collect context
Measurement Context	Measurement Time	Time of collection of context information
	Sensor Location	Location of sensor when context information is collected
	Information Entity Location	Location of the real world entity about which context is collected at the time of collection of context
	Available Attributes	Number of attributes that have a value for that context object
Specifications and Consumer Requirements	Validity Time	Maximum length of time for which a specific type of context information is stable
	Required Attributes	Number of attributes that are required to have a value for that type of context information
	Critical Value	Level of importance of context information of a specific type
	Access Level	Information about the rights of context consumer to access certain type of information
QoC Metric	Reliability	Indicates the extent to which context can be considered credible
	Timeliness	Indicates validity of context to use considering its freshness
	Completeness	All aspects of phenomenon in the environment have been shown
	Significance	Critical value of context information for a specific application
	Usability	Indicates suitability of for use for an intended purpose
	Access Right	Indicate the extent to which owner of context allows the context consumer to access context
	Representation Consistency	Extent to which context representation format is consistent to consumer requirements

Table 1 Brief description of concepts in QoC processing model

4.2 Measurement Context

Measurement context will show the information related to the situation of a specific measurement. This information will include the *Measurement Time, Sensor Location, Information Entity Location and Available Attributes* for a specific type of context object. Table 1 presents the brief description of contextual characteristics. *Measurement Time* is the time at which context information is measured. *Source Location* is the geographical location of the source that collects the context object and *Information Entity Location* is the geographical location of the entity that is represented by that context object. *Source Location* along with *Information Entity Location* will be representing the space resolution. They help to decide about the reliability of a sensor to collect that context object. For example, if we have more than one context object, representing

the same entity in the environment, the context object collected by the sensor closest to the entity will get maximum value of reliability, provided that all the sources are collecting the information with the same accuracy. *Access Level* is also an important contextual characteristic of context. The owner of context can set the level of granularity with which context can be shared with other context consumers to protect his / her privacy. For example, location of a person can be expressed at the level of granularity of country, city, street, and building. Owner of the context can set the access level so that information up to the level of current city of his / her location should be shared with other context consumers.

4.3   Specifications and Consumer Requirements

Consumer of context information will specify the information about their requirements about the quality of context information. These requirements will be used with other information to calculate the subjective metrics of QoC. Table 1 shows the brief description of specifications and consumer requirements. *Validity Time* indicates the length of time for which the value of context remains stable and valid. *Validity Time* will have a different value for each type of information. For example, the location of a fast moving vehicle changes very rapidly and has lower value of *Validity Time* as compared to the location of a walking man. Similarly, stable data, such as profile of an agent in collaborative working environment, does not change very often and has higher value of *Stability Time*. *Critical Value* of context information will indicate that this information is crucial in a specific scenario. This concept particularly affects quality of context information in scenarios where it will be used in emergency tasks. For example, in our afore mentioned scenario, context object having information about the people caught in the low lying area of the city will be of high critical value. *Total Attribute* is the total number of attributes for which a context object can have any value.

4.4   QoC Metrics

QoC metrics are derived from the combination of sensor characteristics, contextual characteristics, and input from specifications and consumer requirements. These metrics will include objective as well as subjective metrics. Objective metrics are calculated independent of the requirements of any context consumer and show that the context information is collected free of error and is suitable to use at an instance of time. Objective metrics include *Reliability*, *Timeliness*, and *Completeness* of context information. Subjective metrics are calculated as quality of context information compared to user requirements for use for a specific purpose. Some of these metrics are *Significance*, *Access Right*, and *Representational Consistency*. Table 1 also present the brief description of objective and subjective QoC metrics.

Reliability indicates the belief that we have in the correctness of information in a context object. It is an objective QoC metric and is calculated independent of the context consumer. *Reliability* of a context object is evaluated from the information provided about the sensor that collects that context object such as accuracy of measurement. Reliability of context object is particularly useful in making selection from different source of same context object, as this metric can tell how reliable was a sensor to collect a particular context object. *Completeness* is an objective QoC metric and indicates that all aspect of the situation in environment has been shown and tell us about the quantity of information that is provided by a context object. Lower value of completeness will indicate ambiguous situations and can result in an undesired action by a context-aware system.

Timeliness is an objective quality measure that indicates the degree of freshness of a context object at a given time. As the situation in pervasive environments changes very rapidly, applications using a context object without having any knowledge about timeliness of that context object may take undesired actions that can result in loss of resources and frustration on the part of user. The value of *Timeliness* and hence the validity of context object decrease as the age of

that context object increases. Therefore, the flood data analyst in our scenario can look at the value of *Timeliness* to be sure of the validity of information contained by that context object. This metric can help him / her to more confidently combine the information contained in that context object with the existing information in context store to provide the current situation of flood in the city to teams participating in rescue work. Context objects, having low value of timeliness, may have misleading or wrong information and can be ignored. If we have static information, e.g., rescue worker's profile information saved at the back end, we can set the lifetime of that information infinite so that its age will not affect the value of *Timeliness* and it will always be maximum.

Significance of context information is a subjective QoC metrics and its value is calculated according to the requirements of a context-aware application and the context of the use of context information. *Significance* of context information indicates the worth or the preciousness of context information in a specific situation. Value of *Significance* is of particular importance in scenarios that involve life threatening situations for humans and its value for a context object will increase if that context object contains information which needs immediate response or attention, e.g., in the case of a context-aware flood response activities that adapt according to current situation on the flood site collapse of a building in a low lying area of the city where the water level is high will have high value of *Significance* as this situation needs immediate response. *Usability* of context information depicts how much that piece of context information is suitable for use with the intended purpose. It will consider the level of granularity of collected context information with the required level granularity. For example, an application needs the information about the location of a person at the level of granularity of street of his location. But if a context information service provides information only about the current city of his location, that context information will have lower value of usability. *Access Right* of context information is also a subjective QoC metric and will vary depending upon who is going to access that context information. Similarly *Presentation Consistency* of context information depends on the format in which information is presented by the context producer and the format that context consumer prefers to use.

5   QoC Evaluation

Figure 2 shows the process of the evaluation of QoC metrics. QoC Evaluator gets the sensor characteristics from the sensors. Context consumer mention some of their requirements for the quality of context information in data model provided to QoC Evaluator. QoC evaluator evaluates the QoC metrics and provide them to context consumer with context information. First step to present QoC metric in a usable form to the context consumer is their quantification, i.e., presenting QoC metrics in a numerical form that is understandable for the context consumer. As we will evaluate QoC metrics against context consumer requirements, it will be appropriate to measure QoC metrics as a decimal which can have value in range [0..1]. Maximum value 1 will mean that QoC metric is in complete compliance to the given requirements while the minimum value 0 will mean total nonconformity to requirements. Sources for the evaluation of QoC metrics are classified as sensor characteristics, contextual characteristics, and specifications and consumer requirements. Table 2 shows the possible methods that can be used for the assessment of different QoC sources. Different QoC sources used to evaluate QoC metrics are shown in Table 3. In the remaining section we will describe how QoC sources are used to evaluate different QoC metrics.

Reliability of a context object, i.e., belief in the correctness of information contained by that context object, is calculated on the basis of accuracy with which sensor has collected context information and the distance of sensor from the entity about which that information is collected. Following equation is used to evaluate the reliability of a context object $\mathcal{O}$.

$$Reliability(\mathcal{O}) = \begin{cases} (1 - \frac{d(\mathcal{S},\mathcal{E})}{d_{max}}) * n\frac{A_i}{A_i} & : \quad if \quad d(\mathcal{S},\mathcal{E}) < d_{max} \\ 0 & : \quad otherwise \end{cases} \qquad (1)$$
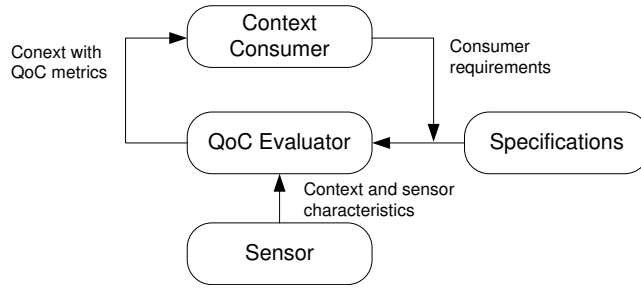
Figure 2 QoC Evaluator evaluating QoC metrics

where $d(\mathcal{S}, \mathcal{E})$ is the distance between the sensor and the entity. d_{max} is the maximum distance for which we can trust on the observation of this sensor. Every type of sensor will have different value for d_{max}. As shown by Equation 1 reliability of a context object is directly proportion to accuracy with which sensor collect context object and is inversely proportion to distance between sensor and entity about which context information is collected.

Classification	QoC Sources	Collection Method / Origin
Sensor Characteristics	Accuracy	Sensor data sampling or sensor specifications
	Granularity	Measurement unit
	Time Period	Sensor configuration
	Sensor State	Sensor configuration
	Sensor Range\Span	Sensor Specification
Measurement Context	Measurement Time	Time stamp at the time of context collection
	Sensor Location	GIS for static sensor and GPS embedded in dynamic sensor
	Information Entity Location	GIS for fixed entities and GPS for dynamic entities
	Available Attributes	Context object
Specification and Consumer Requirements	Validity Time	Context data model
	Total Attributes	Context data model
	Critical Value	Context data model
	Access Level	Context subscription

Table 2 Collection method for QoC sources

Timeliness of a context object can be measured in both objective and subjective ways. To measure the timeliness of a context object independent of consumer requirements we consider the age of context object and time period, i.e., time interval after which sensor takes the new readings. Age of context object $\mathcal{O}$ is calculated by taking the difference between the current time, t_{curr}, and the measurement time of that context object $\mathcal{O}$, $t_{meas}(\mathcal{O})$ as shown by Equation 2.

$$Age(\mathcal{O}) = t_{curr} - t_{meas}(\mathcal{O}) \tag{2}$$

We take the ratio of the age of a context object with the time period for that context object to calculate the objective view of timeliness of that context object. We will also normalize the value of timeliness to have its value in rang [0..1] as shown in Equation 3

$$Timeliness(\mathcal{O}) = \begin{cases} 1 - \frac{Age(\mathcal{O})}{TimePeriod(\mathcal{O})} & : & if \ \ Age(\mathcal{O}) < TimePeriod(\mathcal{O}) \\ 0 & : & otherwise \end{cases} \tag{3}$$

To get the subjective view of the timeliness of a context object $\mathcal{O}$, validity time of context object mentioned by the context consumer is considered in spite of time period as shown in Equation 4.

$$Timeliness(\mathcal{O}) = \begin{cases} 1 - \frac{Age(\mathcal{O})}{ValidityTime(\mathcal{O})} & : & if \ \ Age(\mathcal{O}) < ValidityTime(\mathcal{O}) \\ 0 & : & otherwise \end{cases} \tag{4}$$

The value of timeliness and hence the validity of context object $\mathcal{O}$ decrease as the age of that context object increases and approaches zero when age of context object become equal to time period or validity time of context object considering objective or subjective view of timeliness. Context objects, having low value of the timeliness, may have misleading or wrong information and can be ignored.

QoC Metrics	View		Calculation Method
	Objective	Subjective	
Reliability	X	-	Combination of span reliability and accuracy
Timeliness	X	X	Ratio of age and time period or validity time of context depending upon subjective or objective view
Completeness	X	X	Ratio of available number of attributes to total or required number of attributes of a context object depending upon subjective or objective view
Significance	-	X	Ratio of critical level of context to maximum critical level that type of context can have
Usability	-	X	Comparison of level granularity of context with level of granularity required by context consumer
Access Right	-	X	Comparison of access level of context allowed by context owner to access level of context consumer
Representation Consistency	-	X	Comparison of representation formats

Table 3 Evaluation criteria for QoC metrics

Completeness indicates the quantity of information that is provided by a context object and is calculated as ratio of the sum of weights of available attributes of a context object to the sum of weights of total number of attributes that are required for that context object. As the total number of attributes are provided by a context consumer through context data model, completeness provides the subjective view of QoC. Lower value of completeness will indicate the ambiguous situation and can result in a undesired action by a context-aware system. Completeness, $\mathcal{C}(\mathcal{O})$, of context object $\mathcal{O}$ is evaluated by Equation 5.

$$Completeness(\mathcal{O}) = \frac{\sum_{j=0}^{m} w_j(\mathcal{O})}{\sum_{i=0}^{n} w_i(\mathcal{O})} \qquad (5)$$

In Equation 5 m is the number of available attributes and n is the total or required number of attributes for context object $\mathcal{O}$. As different attributes can have different worth we have used their weights.

Significance indicates the worth or the preciousness of context information in a specific situation and is evaluated considering the critical value of a context object to the maximum critical value that a context object of that type can have. As the levels of critical value are provided by context consumer, significance provides the subjective view of QoC. Significance of context object $\mathcal{O}$ is evaluated by the following equation.

$$Significance(\mathcal{O}) = \frac{CV(\mathcal{O})}{CV_{max}(\mathcal{O})} \qquad (6)$$

where $CV(\mathcal{O})$ is the critical value of the context object $\mathcal{O}$. $CV_{max}(\mathcal{O})$ is the maximum critical value that can be assigned to a context object of the type that is represented by $\mathcal{O}$.

Usability of context information will be measured by comparing the granularity of context information presented by the context object and the granularity of context object that is required by a context consumer. Granularity of a context object is assigned a level in the context model considering the detail of information. For example, the location information of a person is assigned the highest level if it gives the information about the country, city, street, building, and room of the current location of the person and will be assigned the lowest granularity level if context object only gives the information about the country of the current location of that person. Following equation shows the rule for the evaluation of this metric.

$$Usability(\mathcal{O}) = \left\{ \begin{array}{lll} 1 & : & if \;\; GranularityLevel(\mathcal{O}) >= GranularityLevel(\mathcal{CR}) \\ 0 & : & otherwise \end{array} \right. \tag{7}$$

As shown by the above equation usability of context object will be equal to 1 if the granularity level of the context information presented by context object $\mathcal{O}$ is greater than the granularity level of of context information requested by context consumer. Otherwise it will be zero.

Access Right of a context object will be calculated by comparing the access level of granularity of context information that is allowed by the owner of context to the access level that is required by the context consumer. Access right of a context object $\mathcal{O}$ is calculated by the following equation.

$$AccessRight(\mathcal{O}) = \left\{ \begin{array}{lll} 1 & : & if \;\; AccessLevel(\mathcal{O}) >= AccessLevel(\mathcal{CR}) \\ 0 & : & otherwise \end{array} \right. \tag{8}$$

As shown by the above equation, access right will be 1, i.e., context consumer will be allowed to access that context object if access level of context object $\mathcal{O}$ allowed by the context owner is greater than or equal to access level requested by context consumer.

Representation Consistency of a context object depends upon the amount of effort that is needed to transform that context object according to the data model presented by context consumer. If similar data formats are used by sensors and context consumer then representation consistency will have maximum value. Otherwise the value of representation consistency decrease with increase in effort needed to transform the context object according to the requirements of context consumer as shown by the following equation.

$$RepresentationConsistency(\mathcal{O}) = \frac{k}{TransformationEffort} \tag{9}$$

As shown by the above equation, representation consistency of a context object $\mathcal{O}$ will be inversely proportional to the effort that is needed to transform that context object according to context consumer requirements. k is a constant that is used to normalize the value of representational consistency in range [0..1] and its value depends upon the maximum and minimum effort that a context consumer perform for the transformation of data.

6 Quality-Aware Context Management Middleware

Figure 3 shows the components of our quality-aware context management middleware corresponding to the conceptual layers of context management system presented in (Baldauf et al. (2007)). QoC Evaluator is used to evaluate the QoC metrics. *QoC Evaluator* receives context objects as XML elements and evaluates QoC metrics for those context objects. QoC metrics are normalized to have values in range [0..1]. Figure 4 shows a context object annotated with QoC metrics. Components used to evaluate and annotate QoC metrics and conflict resolving policies can be used with any system component. QoC metrics are used to resolve conflicts on those layers (Manzoor (2009b)). Guidelines to select QoC metrics to use in different conflict resolving situations provided to the system in context data model. For example, it can be mentioned that the selection among different sources of a particular type of context information should have been done on the basis of the combination of reliability and timeliness based policies.

We developed our prototype as the part of the implementation of the EU project WORKPAD and is based on COSINE (Juszczyk et al. (2009)). Our context information model, to manage the context information in disaster response, was designed as XML schema. Context consumer pass their requirements to *QoC Evaluator* as an XML file. MXQuery (MXQuery (2010)) and KSOAP2 (KSOAP2 (2010)) , having a low memory foot print to be able to run on mobile devices, were used for processing XML data. In this prototype, we dealt with high level context information. The components dealing with low level context, such as context fusion and high level context extractor, were not implemented yet.
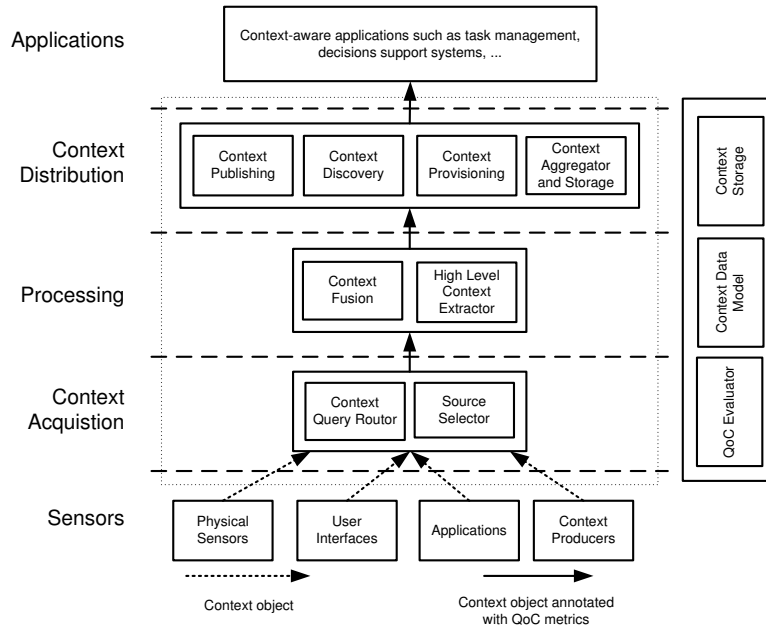
Figure 3 Quality-Aware Context Management Middleware Components

7 Experiments and Evaluation

In our simulated environment, rescue workers were performing their activities to save people in response to flood in a city. Those workers were randomly moving on the flood site and also collecting context about the current situation of flood near an important square in the city. They were sending this context to flood control room where it was combined with existing information to analyze the recent changes on flood site and plan future strategy for rescue activities as described in motivational scenario in Section 2. This context object was of type infrastructure and contained information about the usability of a square in the city. Flood analyst in control room received a lot of context objects with conflicting information and find it difficult to make selection among them. Our context management system evaluated QoC metrics for that context object and annotated that context object with those QoC metrics as shown in Figure 4. Availability of QoC metrics made it possible for the flood analyst to use QoC metrics to perform different tasks while using context information. In the proceeding section we will illustrate how he had used QoC to make decisions while different tasks in context management. While using those QoC metrics he had not considered the fact whether those policies had already been applied to the underlying layers or not.

```
<Infrastructure sourceID = "UIAX00065"
 entityID = "Square000X38"
 timestamp = "1219668617937"
 name = "MainSquare"
 location = "CityCenter"
 Usability = "70%">
<QoCMetrics>
 <timeliness>0.83</timeliness>
 <reliability>0.63</reliability>
 <completeness>0.94</completeness>
 <significance>1.0</significance>
</QoCMetrics>
</Infrastructure>
```
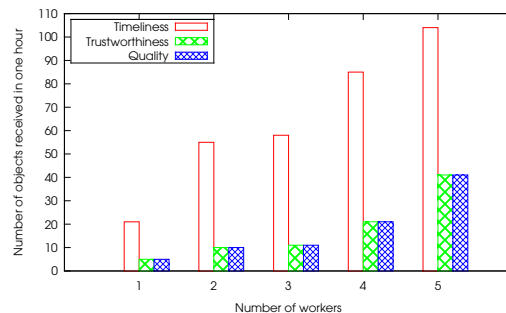

Figure 4 XML representation of context object of type Infrastructure

Figure 5 Source selection using different QoC-based conflict resolving policies

In the first case flood analyst applied QoC metrics at the context acquisition layer to resolve conflict in making selection among different sources that were sending aforementioned context objects. He used reliability, timeliness, and a combination of both reliability and timeliness alternatively to find the best solution. He specified the threshold value for QoC metrics as 0.9. Thus, all the sources of context information that were producing the context objects that had the value of QoC metrics more than threshold had been selected. First, he started by selecting context object on the basis of timeliness only. As in our simulated environment every source of context information was generating context objects after a fixed interval of one minute, the number of context objects having value of timeliness more than specified value increased with increase in sources of context objects. Consequently flood analyst found that only timeliness of context object did not seem to be useful in making source selection in this case as shown in Figure 5. He also checked the results by using reliability and then later combined both reliability and timeliness. He received fewer and more reliable context objects that had been recently collected with a combination of timeliness and reliability and was more satisfied by the results. Figure 5 depicts the number of context objects received by flood analyst in 60 minutes from the selected sources of context information with increase in number of workers using different QoC metrics.

In the second case flood analyst used QoC metrics to enhance the performance of context aggregator so that it contains only useful information and delete all irrelevant and outdated information. In this experiment flood analyst was receiving context objects from five workers that were sending context after random intervals of less than one minute. Flood analyst was initiating a cleaning service after every minute and conflicting context objects that did not meet the specified level of quality were deleted. He used the quality standards based on reliability, timeliness, and a combination of reliability and timeliness and set the threshold value at 0.85. Figure 6 shows the number of context objects that were currently stored in context store using aforementioned policies and threshold value. As it is apparent from Figure 6, using QoC metric reliability did not prove to be very useful as some context objects that have been captured long time ago still have higher value of reliability and are uselessly kept in the context store. Using QoC metric timeliness proved to be same as keeping latest context objects and it has deleted the old context objects that can result in loss of some important context information. Finally, flood analyst used a quality policy based on the combination of both reliability and timeliness to detect useless context objects. With this policy he had not only been able to detect more number of useless context objects but also kept the context objects of high reliability that were very effective in making any decision on the basis of context information.
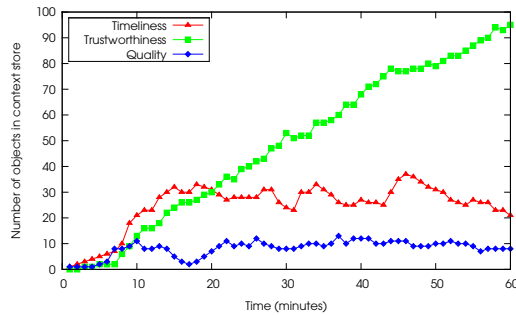
Figure 6 Context aggregation using different QoC-based conflict resolving policies
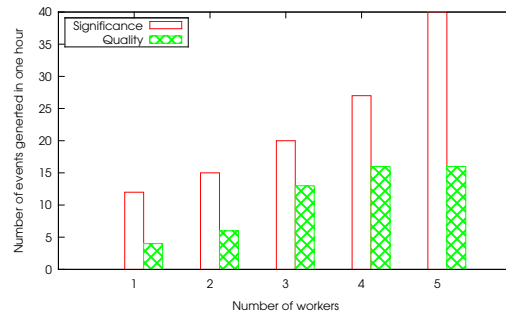
Figure 7 Events generated with help of QoC-based policies

In the final case flood analyst use QoC metrics at the context distribution layer to get notification of specific events on the flood site at it was nearly impossible for him to analyze every context object. Firstly, he only used the significance of context information to generate the events of interest as shown in Figure 7. He observed that only considering the significance of context objects is not sufficient to generate the events of interest. Therefore, he also combined the

reliability of the source of context information with significance and found it quite useful to assist decision making in performing functions at context distribution. In our simulated experiments different QoC metrics were used to perform various task in a middleware to provide context information. QoC metrics have been used individually and in combination with other QoC metrics in different scenarios. The overall behavior of experiments showed that any QoC metric alone was not sufficient to perform the task of decision making on any layer of context-aware systems. QoC metrics used in combination of two or more metrics were more effective to perform this functionality at different layers of context-aware systems. We had also observed that the value of QoC metrics for different applications merely depends on the need of that specific application.

8   Related Work

Many research efforts have been undertaken to design context models that consider the imperfection of context information and strive to present and associate it with context information (Schimidt et al. (2000), Castro et al.(2000), Gray et al. (2001)). Some works have also used metadata to indicate the characteristics of context information and discussed the advantages of presenting metadata with context information (Lei et al. (2002), Henricksen et al. (2004), Hönle et al. (2005)). QoC was first defined in (Buchholz et al. (2003)) and then in (Kraüse et al. (2005)) as discussed in Section 3. Both of these definitions consider only objective view of quality of context information as being free of error. In our definitions we have considered both objective and subjective view of QoC to emphasize that context information is not only free of errors but also reveal that how much context information conforms to the requirements of a context consumer. Research efforts have also emphasized the importance of QoC and indicated QoC metrics considering temporal characteristics and correctness of context information, amount of information contained by a context object, and observation level and trust on sensors (Gray et al. (2001), Buchholz et al.(2003), Henricksen et al. (2004), Hönle et al. (2005)). Villalonga et al. have also analyzed QoC metrics present in research literature and presented the guidelines to define QoC metrics in activity recognition domain (Villalonga (2009)). As compared to our work these work had considered only the objective nature of QoC metrics and defined QoC metrics independent of any requirements from the context consumer. We have introduced some new QoC metrics such as significance of context information that is particularly useful in critical situations. Batini et al. have provided a detailed survey of metrics that are defined in data quality domain (Batini et al. (2009)). Those metrics are more suitable for persistent data as compared to context that is volatile data and changes very rapidly.

QoC Evaluation: Though QoC metrics have been indicated few works have tried to evaluate these metrics. In (Schmidt (2005)), incompleteness, inconsistency and variation in precision of context information have been identified as sources of imperfection in context information. However, only the age of context information has been used to measure the confidence in context information. An analysis of QoC indicators, such as precision, freshness, spatial resolution, temporal resolution, and probability of correctness and different options to quantify these metrics has been presented in (Sheikh (2008)). But no mechanism has been provided to evaluate these metrics. A relationship between dimensions of information quality and QoC metrics have been discussed in (Kim et al. (2006)). This work used a statistical estimation method to calculate the accuracy of sensor data in smart homes. However, their method to measure accuracy is only appropriate in those cases where sensors get continuous data around some average value, e.g., via temperature sensors. Completeness is also measured as the ratio of available attributes to total number of attributes for a specific context object. As compared to our work, they have also not provided the evaluation of enough QoC metrics to be useful with the applications in pervasive computing environments.

QoC in Context-Aware Systems: In (Bu et al. (2006)), they used the conflict resolving policy to delete the conflicting objects with smaller value of a measure, relative frequency, that is based on the time of the generation of that context object. In our work, we have used more

sophisticated policies based on QoC metrics to resolve the conflict among context objects. In (Schmidt (2005)), a model had used the age of context information to decide about the validity of the context information. However, age of context information is not enough for context data management in more dynamic pervasive environments. QoC has also been used in tasks, such as, context selection (Herbscher et al.(2006), Chantzara et al.(2006), Pawar et al.(2006), Breza et al.(2007)), inconsistency resolution (Bu et al. (2006)), privacy enforcement (Neisse et al(2007), Sheikh et al. (2008)), context aggregation (Manzoor (2009a)), and context reasoning(McKeever (2009)). But most of these works have used one or two metric and have not exploited QoC to full extent.

9 Conclusion and Future Work

In this paper we have defined QoC by considering its multi facetted nature. We presented a model for processing QoC metrics. We have also presented our middleware that passes QoC metrics along with context information to context consumer. We have performed the experiments to evaluate the importance of different QoC metrics in performing tasks in a context-aware middleware. We found that QoC metrics used in combination with each other that are selected considering the nature of task can be more effective than an individual QoC metrics. For our next steps, we plan to use these QoC metrics to do more sophisticated reasoning in the fusion of low level context and extraction of high level context information. We also plan to enhance the quality of context information by combining the context information and QoC metrics from more than one context objects. Further we plan to study the interdependence of QoC, Quality of Data, and Quality of Service and enhance the evaluation of our QoC metrics.

References

Baldauf, M., Dustdar, S. & Rosenberg, F. 2007 A survey on context-aware systems. International Journal of Ad Hoc and Ubiquitous Computing. Volume 2(4), Pages 263-277, Inderscience Publishers.

Batini, C., Cappiello, C., Francalanci, C. & Maurino, A. 2009 Methodologies for data quality assessment and improvement. ACM Computing Surveys (CSUR) Volume 41(3), Pages 1-52, ACM Publishers.

Breza, M., Anthony, R., & McCann, J. 2007 Quality of context driven autonomicity. In 2nd International Workshop on Engineering Emergence in Decentralised Autonomic Systems. IEEE Computer Society.

Bu, Y., Gu, T., Tao, X., Li, J., Chen, S. & Lu, J. 2006 Managing quality of context in pervasive computing. In QSIC 2006: Proceedings of the Sixth International Conference on Quality Software, Pages 193200. IEEE Computer Society.

Buchholz, T., Küpper, A., & Schiffers, M. 2003 Quality of Context: What it is and Why ee need it. In Proceedings of the 10th International Workshop on the HP OpenView University Association (HPOVUA), Springer.

Castro, P. and Muntz, R. 2000 Managing context data for smart spaces. IEEE Personal Communications, IEEE [see also IEEE Wireless Communications], Volume 7(5), Pages 44-46, IEEE Computer Society.

Chantzara, M., Anagnostou, M. & Sykas, E. 2006 Designing a quality-aware discovery mechanism for acquiring context information. In AINA 2006: Proceedings of the 20th International Conference on Advanced Information Networking and Applications. Volum 1, Pages 211216, IEEE Computer Society.

Chen, G. & Kotz, D. 2002 Context Aggregation and Dissemination in Ubiquitous Computing Systems, Proceedings of the Fourth IEEE Workshop on Mobile Computing Systems and Applications, IEEE Comuter Society.

Dey, A. K. 2001 Understanding and Using Context, Personal and Ubiquitous Computing Journal, Volume 5, Pages 4-7, Springer.

Dorn, C., Schall, D., & Dustdar, S. 2006 Granular Context in Collaborative Mobile Environments. In On the Move to Meaningful Internet Systems, Proceedings of OTM Workshops 2006, Pages 1904 - 1913, Springer.

Formayer, H. & Frischauf, C. 2004 Extremereignisse und klimawandel in sterreich aus sicht der forschung. Technical report, WWF Austria and Institute of Meteorology, BOKU, Vienna.

Gray, P. & Salber D. 2001 Modelling and Using Context Information in the Design of Interactive Applications. Engineering for Human-Computer Interaction. Volume 2254/2001, Pages 317-335, Springer.

Henricksen, K., Indulska, J. & Rakotonirainy, A. 2002 Modeling Context Information in Pervasive Computing Systems. Proceedings of the First IEEE Annual Conference on Pervasive Computing and Communications Workshops, Pages 79-117. Springer.

Henricksen, K. & Indulska, J. 2004 Modeling and Using Imperfect Context Information. In PER-COMW'04: Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, IEEE Computer Society.

Henricksen, K., Indulska, J., McFadden, T. & Balasubramaniam, S. 2005 Middleware for Distributed Context-Aware Systems In Proceedings of International Symposium on Distributed Objects and Applications, Springer.

Hönle, N., Kappeler, U. P., Nicklas, D., Schwarz, T., & Grossmann, M. 2005 Benefits of integrating meta data into a context model. In Mobility Aware Technologies and Applications, volume 3744, Pages 324 333, Springer.

Huebscher, M.C., McCann, J.A. & Dulay, N. 2006 Fusing multiple sources of context data of the same context type. In Proceedings of the 2006 International Conference on Hybrid Information Technology, IEEE Computer Society.

Juran, J. M., Godfrey, A. B., Hoogstoel, R. E. & Schilling, E. G. 1998 Quality Control Handbook. McGraw-Hill.

Juszczyk, L., Psaier, H., Manzoor, A. & Dustdar, S. 2009 Adaptive query routing on distributed context - the cosine framework. In Proceedings of International Workshop on the Role of Services, Ontologies, and Context in Mobile Environments (ROSOC-M) 10th International Conference on Mobile Data Management (MDM09) Taipeh, Taiwan. IEEE Computer Society.

Kim, Y. & Lee, K. 2006 A quality measurement method of context information in ubiquitous environments. In ICHIT 06: Proceedings of the 2006 International Conference on Hybrid Information Technology, pages 576581, IEEE Computer Society.

Kraüse, M. & Hochstatter, I. 2005 Challenges in modeling and using Quality of Context(QoC). In Proceedings of Mobility Aware Technologies and Applicaitons, volume 3744/2005, pages 324-333. Springer.

KSOAP2. http://sourceforge.net/projects/ksoap2/. Access Date: May 1, 2010.

Lei, H., Sow, D. M., Davis, J. S. II., Banavar, G., & Ebling, M. R. 2002 The design and applications of a context service. SIGMOBILE Mob. Comput. Commun. Rev., 6(4):45-55.

Manzoor, A., Truong, H. L. & Dustdar, S. 2008 On the evaluation of quality of context. In Proceedings of Third European Conference on Smart Sensing and Context, EuroSSC, volume 5279, pages 140153. Springer.

Manzoor, A., Truong, H. L. & Dustdar, S. 2009a Quality Aware Context Information Aggregation System for Pervasive Environments. 2009 International Conference on Advanced Information Networking and Applications Workshops, Pages: 266-271, IEEE Computer Society.

Manzoor, A., Truong, H. L. & Dustdar, S. 2009b Using Quality of Context to Resolve Conflicts in Context-Aware Systems. Proceedings of First International Conference on Quality of Context. Springer.

Mostefaoui, G.K., Pasquier-Rocha, J. & Brezillon, P. 2004 Context-Aware Computing: A Guide for the Pervasive Computing Community. IEEE/ACS International Conference on Pervasive Services Pages 39-48, IEEE Computer Society.

Neisse, R. , Wegdam, M., van Sinderen, M., & Lenzini, G. 2007 Trust management model and architecture for context-aware service platforms. In OTM Conferences (2), volume 4804 of Lecture Notes in Computer Science, pages 18031820. Springer.

Pawar, P. & Tokmakoff, A. 2006 Ontology-Based Context-Aware Service Discovery for Pervasive Environments. In 1st IEEE International Workshop on Services Integration in Pervasive Environments (SIPE 2006), Lyon, France. Pages 1-7. IEEE Computer Society.

MXQuery. http://mxquery.org/. Access Date: May 1, 2010.

Ranganathan, A., Al-Muhtadi, J., & Campbell, R. H. 2004 Reasoning about Uncertain Contexts in Pervasive Computing Environments. Pervasive Computing, Volume 3(2), IEEE Computer Socienty.

Schmidt, A. 2000 Implicit human computer interaction through context. Personal and Ubiquitous Computing, 4(2/3), Springer.

Sheikh, K., Wegdam, M. & van Suinderen, M. 2008 Quality-of-context and its use for protecting privacy in context aware systems. Journal of Software Volume 3, Pages 8393, Academy Publishers.

Schmidt, A. 2005 A layered model for user context management with controlled aging and imperfection handling. In Modeling and Retrieval of Context. Proceedings of the 2nd International Workshop on Modeling and Retrieval of Context, Springer.

The EU WORKPAD Project. 2007-2009 http://www.workpad-project.eu.

Truong, H. -L., Juszczyk, L., Manzoor, A., & Dustdar, S. 2007 Escape- an adaptive framework for managing and providing context information in emergency situation. In Smart Sensing, and Context, volume 4793/2007, pages 207-222. Springer.

Villalonga, C., Roggen, D., Lombriser, C., Zappi, P., & Trster, G. 2009 Bringing Quality of Context into Wearable Human Activity Recognition Systems. Proceedings of First International Conference on Quality of Context. Springer.

Weiser, M. 1991 The Computer for the 21ST Century. SCIENTIFIC AMERICAN, Volume 265, NO.3/SEP, P. 94, 1991