

On the Evaluation of Quality of Context*

Atif Manzoor, Hong-Linh Truong, and Schahram Dustdar

Distributed Systems Group, Vienna University of Technology
{manzoor, truong, dustdar}@infosys.tuwien.ac.at

Abstract. High quality context information plays a vital role in adapting a system to the rapidly changing situations. However, the diversity of the sources of context information and the characteristics of the computing devices strongly impact the quality of context information in pervasive computing environments. Quality of Context parameters can be used to characterize the quality of context information from different aspects. In this paper, we quantify the Quality of Context parameters to present them in a suitable form for use with the applications in pervasive environments. We also present a mechanism to tailor the Quality of Context parameters according to the needs of an application and then evaluate these parameters. Enrichment of context information with Quality of Context parameters enhances the capabilities of context-aware systems to effectively use the context information to adapt to the emerging situations in pervasive computing environments.

1 Introduction

Designing an adaptive system in pervasive computing environments faces a stern challenge because of the limitations of the quality of context information [2]. Context information has an innate characteristic of imperfection [4] and its quality is highly influenced by the way it is acquired in pervasive computing environments [28]. Diverse sources of context information, ranging from physical and logical sensors to user interfaces and applications on mobile devices, also affect the quality of the context information and context information can be imperfect [13].

Many research efforts have been done to provide and use context information to achieve the vision of pervasive computing environments since it has been presented in [29]. However, as indicated in [1], existing systems rarely give any consideration to the quality of context information. Previous studies show that the applications using context information either make wrong assumptions about its quality [5, 7, 13] or perform extra effort to deal with the uncertainty of context information [20].

As discussed in [14], “context information is (a) incorrect if it fails to reflect the true state of the world it models, (b) inconsistent, e.g., by containing contradictory information, or (c) incomplete if some aspects of context are unknown”. Therefore, possessing the knowledge about the quality of the context information plays an important role in using that information effectively and achieving the capability of context awareness.

* This research is partially supported by the European Union through the FP6-2005-IST-5-034749 project WORKPAD.

Subsequently, research efforts have been undertaken to model the quality of context information [12, 22]. The term “Quality of Context(QoC)” has been coined in [3] and is defined as “any information that describes the quality of information that is used as context information”. QoC parameters have also been listed as the probability of correctness, trust-worthiness, resolution, and up-to-dateness [3]. However, few works [16, 23] have discussed the evaluation of QoC parameters. There has not been any work which presents and evaluates the QoC parameters as the worth of context information for an application and provides the context information enriched with these QoC parameters.

In this paper we evaluate the QoC and relate QoC to the worth of context information for an application. We classify QoC into QoC sources and parameters. QoC sources are the information about the sources that collect context information, the environments where that context information is collected, and the entities about which the context information is collected. These QoC sources are used to evaluate QoC parameters that indicate the worth of context information for an application. Context information enriched with QoC parameters is provided to the applications to make them aware of the quality of context information they use. We also discuss our context information model that represents the context information enriched with QoC parameters. We have implemented our approach to provide context information enriched with QoC parameters to support adaptiveness in disaster response activities.

The rest of paper is organized as follows: Section 2 illustrates a scenario and discusses the motivation for our work. Section 3 gives an overview of related work and compares them with our approach. Section 4 presents a classification of QoC sources and QoC parameters. Our approach to evaluate the QoC parameters is presented in Section 5. We discuss our context information model enriched with QoC parameters in Section 6. Our prototype implementation is explained in Section 7. Finally, we conclude the paper and discuss future work in Section 8.

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 the 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. Losses of human life and livestock, damages on the infrastructure, buildings, public and private properties rose the public awareness and the demand for 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 the floods.

This scenario is illustrated in [11] and is supported in the EU project WORKPAD [25]. In this project, we aim at facilitating people, performing rescue work in such situations, by providing context information to them. Field workers, participating in the rescue activities and equipped with mobile devices, share context information among themselves and send it to the back end by using context information management services as described in our ESCAPE framework [27]. But due to the unawareness about

the quality of information, people face problems in using this information. For example, in afore mentioned scenario, a geographical information specialist makes an analysis of the damage caused by the flood and provides this information to the organizations participating in the rescue activities. He receives context information from the field workers and combines it with the data stored in a geographical information system (GIS) 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 information to be aware of the quality of that information. The photos taken all look the same and he has often not been able to select one of them and relate it to the digital map stored in the GIS. These problems not only cost him time and effort, but also affect the quality of his work. Context information, enriched with QoC parameters such as up-to-dateness, trust-worthiness, completeness, and significance, allows him to know the quality of context information without looking at the contents of context information, and thus substantially improving his work.

3 Related Work

Quality of context information, gathered in pervasive computing environments, has been considered unsatisfactory since the start of research in context-aware systems [7, 14]. It has also been considered a challenge to deal with this shortcoming of the quality of context information in the development of context-aware systems [8, 12]. But only few works [14, 8] have used metadata to indicate characteristics of context information and advantages of modeling metadata with context information, e.g., [15]. Subsequently, problems associated with this imperfection of context information and its causes have been explored in more detail [13]. Here, we discuss the works that defined the term Quality of Context, identified the parameters to indicate the Quality of Context, and evaluated some of them.

3.1 Quality of Context

Quality of Context (QoC) was first defined in [3] as “any information that describes the quality of information that is used as context information”. Later on, QoC has also been defined in [17] as “any inherent information that describes context information and can be used to determine the worth of information for a specific application”. Precision, probability of correctness, trust-worthiness, resolution, and up-to-dateness has been identified as important QoC parameters in [3]. This list has been extended to include accuracy, completeness, representation consistency, and access security selected on the basis of user’s concern in the quality of context information [16]. The characteristics of the sensor, the situation of the specific measurement, the values expressed by the context information object itself, and the granularity of the representation format have also been identified as the sources to determine QoC [17].

However, these works have not provided the QoC parameters in a form that can show the worth of context information for an application and to allow those parameters to be used by an application. They have also not made any distinction between the QoC indicators which can be used to calculate high level QoC parameters for application usage.

3.2 QoC Evaluation

Though QoC parameters have been indicated and context information models have been designed to accommodate these parameters with context information [12, 21, 24, 30], few works have tried to evaluate these parameters. In [22], incompleteness, inconsistency and variation in precision of context information have been identified as sources of imperfection in context information. A three-layered model, for user context management, has been discussed to handle this problem of imperfect context information. However, only the age of context information has been used to measure the confidence in context information that is not enough to indicate the quality of context information in pervasive computing environments.

An analysis of QoC indicators, such as precision, freshness, spatial resolution, temporal resolution, and probability of correctness and different options to quantify these parameters, has been done in [23]. Later, this work has used these QoC parameters to enforce the user privacy policy in a health tele-monitoring scenario. But no mechanism has been provided to evaluate these parameters for context information and provide these parameters to the applications. As compared to their work, we have also related QoC parameters to the worth of context information for an application.

A relationship between dimensions of information quality and QoC parameters have been discussed in [16]. Accuracy, completeness, representation consistency, access security, and up-to-dateness have been selected as quality dimensions of context information. 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. This work has also not provided the evaluation of enough QoC parameters to be useful with the applications in pervasive computing environments.

4 QoC Classification

As we aim at relating QoC to the worth of context information for an application, we classify QoC into QoC sources and QoC parameters. QoC sources are the quantities which are sensed from the environment or profiled as the configuration of a system. But as these values are not appropriate for use with an application, they are transformed to higher level QoC parameters that are in more suitable form for computational use. Context information is then annotated with these QoC parameters and is provided to users or applications.

Let \mathcal{O} be a context object that has context information about a real world entity \mathcal{E} . The context object \mathcal{O} is collected by a sensor \mathcal{S} . Sensor \mathcal{S} can be a physical sensor or a logical sensor, e.g., user interfaces and applications on a portable device.

4.1 QoC Sources

QoC sources describe the information about the source of context information, environment in which context information is gathered and a context information object itself.

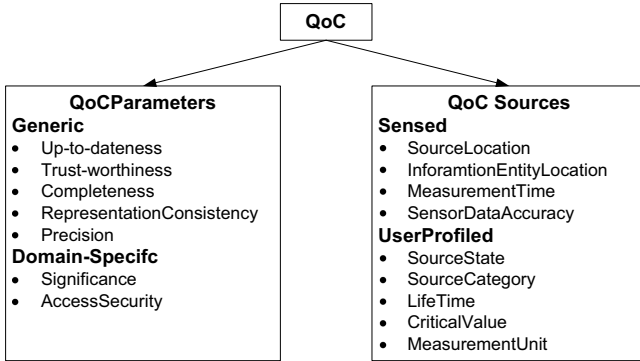


Fig. 1. Classification of QoC into sources and parameters

They include information that have been sensed from the environment, derived from existing information, or profiled at the source of context information. These values will be used to determine the QoC parameters. Figure 1 shows the QoC sources which are collected along with context object. Here we will give a short description of each of them.

SourceLocation is the geographical location of the source that collects the context object and *InformationEntityLocation* is the geographical location of the entity that is represented by that context object. *SourceLocation* along with *InformationEntityLocation* will be representing the space resolution. They help to decide about the trust-worthiness on the context object. For example, if we have more than one context object, representing the same entity in the environment, the context object reported by the source that is nearest the entity will get maximum value of trust-worthiness, provided that all the sources are collecting the information with the same accuracy.

MeasurementTime is the time at which context information is measured. *SensorDataAccuracy* is the accuracy with which a sensor can collect a context object. *SourceState* 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 *SourceState*. While sensors embedded in a portable device carried by a human have dynamic value for the *SourceState*, e.g., GPS sensors embedded in mobile phones. *SourceCategory* indicates category that a source of context information can have in pervasive computing environments. Possible values for the *SourceCategory* can be sensed, profiled, derived, and static [13]. *LifeTime* is the period of time after which context information becomes obsolete and it is necessary to take its value again. For example, the location of fast moving vehicle may has less value regarding lifetime as compared to the location of a walking person.

CriticalValue 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 tasks involving saving the human lives. For example, in our afore mentioned scenario, context object having information about the people caught in the low lying area of the city will by of high critical value. *MeasurementUnit* is used to describe precision of context information. The location of an entity measured

Table 1. Association between the QoC parameter and sources

<i>QoC parameter</i>	<i>QoC sources used for evaluation</i>
Up-to-dateness	MeasurementTime, CurrentTime
trust-worthiness	SourceLocation, InformationEntityLocation, SensorDataAccuracy
completeness	Ratio of number of attributes filled to total number of attributes
significance	CriticalValue

with the accuracy of ten meters will have less precision as compared to a measurement up to the accuracy of one meter.

4.2 QoC Parameters

QoC parameters are derived from QoC sources and are represented in a form that is suitable for use by an application. Figure 1 shows QoC parameters divided into generic and domain specific parameters. Generic QoC parameters are those parameters which are required by most applications, such as precision, trust, validity, representation consistency, and completeness [3]. Domain specific QoC parameters are those parameters that are important for some specific application domains, such as significance and access security. Generic QoC parameters have been described in different works [3, 17, 23]. Significance of context information is important in life threatening situations such as in emergency response. It signals unusual events, in the environment, which need immediate attention. For example, in a flood response scenario if water level raises more than threshold value, this information will be of high significance and will need immediate response. In the next section we will discuss how can we quantify and evaluate some of the QoC parameters.

5 QoC Evaluation

In this section we discuss our approach and system that evaluates QoC parameters. The first step to represent QoC parameters in such a form that they can easily be used by an application is to quantify them. Quality is a relative term and is measured against some standards so it is appropriate to measure QoC parameters as a decimal which can have value in the range $[0..1]$. Maximum value 1 means that QoC parameter is in complete compliance to the given requirements while minimum value 0 means total nonconformity to the requirements. Table 1 shows the QoC parameters which are evaluated in this section and QoC sources which are used for their evaluation. We have selected those QoC parameters for close examination which are more expressive and have substantial impact on our illustrated scenario.

5.1 Up-To-Dateness

This quality measure indicates the degree of rationalism to use a context object for a specific application at a given time. We take into account the age of context object and the lifetime of that context object to calculate the value of up-to-dateness. The age of the

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 1.

$$Age(\mathcal{O}) = t_{curr} - t_{meas}(\mathcal{O}) \quad (1)$$

Up-to-dateness of the context object \mathcal{O} , $\mathcal{U}(\mathcal{O})$, is calculated by Equation 2.

$$\mathcal{U}(\mathcal{O}) = \begin{cases} 1 - \frac{Age(\mathcal{O})}{Lifetime(\mathcal{O})} & : \text{if } Age(\mathcal{O}) < Lifetime(\mathcal{O}) \\ 0 & : \text{otherwise} \end{cases} \quad (2)$$

The value of up-to-dateness and hence the validity of context object \mathcal{O} decrease as the age of that context object increases. Therefore, the geographical information specialist in our scenario can look at the value of this QoC parameter to be sure of the validity of information contained by the context object. This parameter can help him to more confidently combine the information contained in this context object with the existing information in GIS to provide the current situation of flood in the city to teams participating in rescue work. Context objects, having low value of the up-to-dateness, 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 up-to-dateness and it will always be maximum.

5.2 Trust-Worthiness

This quality measure indicates the belief that we have in the correctness of information in a context object. Trust-worthiness of a context object is highly affected by the space resolution, i.e., the distance between the sensor and the entity. The farther the sensor from the entity the more will be the doubt in the correctness of information presented by that context object. Along with the space resolution, accuracy, with which the sensor collects a context object, also affects the trust-worthiness of that context object. Let the accuracy of the sensor data be δ . The trust-worthiness, $\mathcal{T}(\mathcal{O})$, of context object \mathcal{O} is defined by Equation 3.

$$\mathcal{T}(\mathcal{O}) = \begin{cases} (1 - \frac{d(S, \mathcal{E})}{d_{max}}) * \delta & : \text{if } d(S, \mathcal{E}) < d_{max} \\ 0 & : \text{otherwise} \end{cases} \quad (3)$$

where $d(S, \mathcal{E})$ 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} . For example d_{max} value for a satellite capturing the photos from the space will be a lot more than the camera held by a work to take photos in the field. Accuracy of a sensor, δ , is measured on the basis of a statistical estimation method presented in [16].

Trust-worthiness is useful in situations when we have more than one context object representing the same entity. There will be more confidence in the context object collected by the sensor that has a higher value of trust-worthiness. For instance, in scenario discussed in Section 2, photos will be preferred from a worker that have higher value of trust-worthiness. It means that the worker is nearest the entity, represented by the photo, and he has device capability to capture the photo with more accuracy.

5.3 Completeness

This quality measure indicates the quantity of information that is provided by a context object. In [16] completeness has been computed as the ratio of the number of attributes available to the total number of attributes. We have enhanced this concept by using the weights for different attributes, as all attributes of a context object do not have the same importance. Hence, we define the completeness of a context object as the ratio of the sum of the weights of available attributes of a context object to the sum of the weights of all the attributes of that context object. Completeness, $\mathcal{C}(\mathcal{O})$, of context object \mathcal{O} is evaluated by Equation 4.

$$\mathcal{C}(\mathcal{O}) = \frac{\sum_{j=0}^m w_j(\mathcal{O})}{\sum_{i=0}^n w_i(\mathcal{O})} \quad (4)$$

where m is the number of the attributes of context object \mathcal{O} that have been assigned a value and $w_j(\mathcal{O})$ represents the weight of the j th attribute of \mathcal{O} that has been assigned a value. Similarly, n is the total number of the attributes of context object \mathcal{O} and $w_i(\mathcal{O})$ represents the weight of the i th attribute of \mathcal{O} . The value of completeness will be maximum, i.e., 1 if $n = m$. It means that all the attributes of context object \mathcal{O} have been assigned a value.

5.4 Significance

This quality measure indicates the worth or the preciousness of context information in a specific situation. Its value is of particular importance in scenarios that involve life threatening situations for humans. For example, in the flood scenario described in Section 2, significance of the context object will increase if it contains information which needs immediate response or attention, e.g., collapse of a building in a low lying area of the city where the water level is high. A context object having this information gets a higher value of significance so that it will get immediate attention. Significance of context object \mathcal{O} , $\mathcal{S}(\mathcal{O})$, is evaluated by Equation 5.

$$\mathcal{S}(\mathcal{O}) = \frac{CV(\mathcal{O})}{CV_{max}(\mathcal{O})} \quad (5)$$

where $CV(\mathcal{O})$ is the critical value of the context object \mathcal{O} . This information will be gathered from a situation configuration file. This configuration file will have the information about the critical values of each type of concept in the context model for a specific situation. $CV_{max}(\mathcal{O})$ is the maximum critical value that can be assigned to a context object of the type that is represented by \mathcal{O} .

5.5 Algorithm for the Evaluation of QoC Parameters

Algorithm 1 has been defined to evaluate the QoC parameters. This algorithm takes a context object \mathcal{O} as input and requires that the QoC source information is attached to the context object. It calculates the values of QoC parameters and returns the context object after annotating it with these parameters.

Algorithm 1. Evaluation of QoC parametersINPUT: ContextObject \mathcal{O} with QoC sourcesOUTPUT: ContextObject \mathcal{O} with QoC parameters

```

1:  $qocParameters \leftarrow null$ 
2:  $qocSources \leftarrow get\ QoCSources\ from\ \mathcal{O}$ 
3: if  $qocSources \neq null$  then
4:    $t_{meas}(\mathcal{O}) \leftarrow get\ MeasurementTime\ of\ \mathcal{O}$ 
5:    $age \leftarrow t_{curr} - t_{meas}(\mathcal{O})$ 
6:    $lifetime \leftarrow get\ LifeTime\ of\ \mathcal{O}$ 
7:   if  $age < lifetime$  then
8:      $up - to - dateness \leftarrow 1 - (age/lifetime)$ 
9:   end if
10:  {set value of validity in qocParameters}
11:   $qocParameters.Up - to - dateness \leftarrow up - to - dateness$ 
12:   $loc_s \leftarrow get\ SourceLocation\ from\ \mathcal{O}$ 
13:   $loc_e \leftarrow get\ InformationEntityLocaiton\ from\ \mathcal{O}$ 
14:   $distance \leftarrow distance\ between\ loc_s\ and\ loc_e$ 
15:   $accuracy \leftarrow get\ SensorDataAccuracy\ from\ \mathcal{O}$ 
16:  if  $distance < distance_{max}$  then
17:     $trust - worthiness \leftarrow (1 - distance/distance_{max}) * acc$ 
18:  end if
19:  {set value of trust-worthiness in qocParameters}
20:   $qocParameter.Trust - worthiness \leftarrow trust - worthiness$ 
21:  for each attribute in  $\mathcal{O}$  do
22:    if value for attribute is available then
23:      {Sum of the weights of attributes which have the value}
24:       $w_a \leftarrow w_a + weight\ assigned\ to\ attribute$ 
25:    end if
26:    {Sum of the weights of all the attributes}
27:     $w_c \leftarrow w_c + weight\ assigned\ to\ attribute$ 
28:  end for
29:   $completeness \leftarrow w_a/w_c$ 
30:  {set value of completeness as qocParameters}
31:   $qocParameter.Completeness \leftarrow completeness$ 
32:   $significance \leftarrow critical\ value\ of\ \mathcal{O}/maximum\ critical\ value\ that\ \mathcal{O}\ can\ have$ 
33:  {set significance as value of qocParameters}
34:   $qocParameters.Significance \leftarrow significance$ 
35:  {Annotation of ContextObject  $\mathcal{O}$  with qocParameters}
36:   $\mathcal{O}.QoCParameters \leftarrow qocParameters$ 
37: end if
38: RETURN  $\mathcal{O}$ 

```

At Line 1 the algorithm extracts QoC source information that is attached to context object \mathcal{O} . If QoC sources information is not provided then it returns the context object without the values of QoC parameters. At Line 1 it gets the measurement time of context object \mathcal{O} that is used to calculate the age of the context object at Line 1. Validity is calculated at Line 1 and is added to QoC parameters at Line 1.

The algorithm gets the location of the sensor that gathers the context object at Line 1 and gets the location of the entity that is represented by the context object at Line 1. The distance between the sensor and the entity is calculated at Line 1 and is used to calculate the trust-worthiness of context object \mathcal{O} at Line 1. Trust-worthiness is assigned to QoC parameters at Line 1. At Line 1 it calculates the sum of the weights of attributes that have been assigned a value while the sum of the weights of all attributes is calculated at Line 1. Completeness is calculated at Line 1 and assigned to QoC parameters at Line 1. Finally the significance of context object \mathcal{O} is calculated at Line 1 and added to QoC parameters at Line 1. Context object \mathcal{O} is enriched with QoC parameters at Line 1 and returned at Line 1.

6 Enrichment of Context Information Model with QoC

Developing a model to represent context information is of foremost importance in a context-aware system. Without a suitable model it is impossible to provide context information. The main considerations to model context information are: What are the system requirements? What is the purpose of collecting information? How is the information going to be used? Who will use this information? Another factor in the provisioning of context information is to deliver the user with only necessary information considering his current situation.

```

<xs:complexType name="QoC-Source">
  <xs:sequence>
    <xs:element name="InformationEntityLocation" type="ci:GeoLocationType"/>
    <xs:element name="Accuracy" type="ci:QoC-Decimal"/>
    <xs:element name="SourceLocation" type="ci:GeoLocationType"/>
  </xs:sequence>
  <xs:attribute name="SourceState" type="ci:SourceStateType" />
  <xs:attribute name="SourceCategory" type="ci:SourceCategoryType"/>
  <xs:attribute name="LifeTime" type="xs:duration"/>
  <xs:attribute name="CriticalValue" type="ci:CV_Integer"/>
  <xs:attribute name="MeasurementUnit" type="xs:string"/>
  <xs:attribute name="MeasurementTime" type="xs:dateTime"/>
</xs:complexType>

```

Fig. 2. (Simplified) XML schema representation of QoC source

To achieve these requirements, our model to provide context information for supporting disaster response is based on W4H [6] and granular context model [9]. The main concepts in this model are the *SupportWorkers* that perform rescue work on disaster site, *Devices* that are used by the *SupportWorkers* for communication and collection of context information, *Activities* of the *SupportWorkers*, and the *Site* of disaster response. Each concept is represented by a context object and can contain other context objects. For example, the context object providing information about a *SupportWorker* includes the context object providing information about the *Device* used by that *SupportWorker*. Each context object also include *QoCSources* information that are described in Section 4 and *QoCParameters* that are evaluated in Section 5.

```

<xs:complexType name="QoCParametersType">
  <xs:sequence>
    <xs:element name="Up-to-dateness" type="ci:QoC-Decimal"/>
    <xs:element name="Trust-worthiness" type="ci:QoC-Decimal"/>
    <xs:element name="Completeness" type="ci:QoC-Decimal"/>
    <xs:element name="Significance" type="ci:QoC-Decimal"/>
  </xs:sequence>
</xs:complexType>
<xs:simpleType name="QoC-Decimal">
  <xs:restriction base="xs:decimal">
    <xs:minInclusive value="0.0"/>
    <xs:maxExclusive value="1.0"/>
  </xs:restriction>
</xs:simpleType>

```

Fig. 3. (Simplified) XML schema representation of QoC-parameters

Figure 2 shows the XML representation of the QoC sources that will be attached to every context object and Figure 3 shows the XML representation of the QoC-parameters. Every QoC-parameter is of type QoC-Decimal which is defined as the restricted type over decimal to have the value in range [0.0, 1.0].

7 Implementation

The components of our system that evaluate the QoC parameters (QoC Evaluator), annotate the context information (QoC Annotator), and provide QoC enriched context information (QoC Provider) to the applications using it are shown in Figure 4. We have implemented this system by using our Vimoware toolkit [26] that supports the development of Web services in ad-hoc networks of mobile devices. Sensors, collecting the information from the environment, are implemented as Context Information Management Service (CIMS). These sensors include the physical and logical sensors such as GPS sensor, battery sensor, memory sensor, network sensor, and sensors embedded in user interfaces and applications on the mobile devices.

CIMS publishes the type of context object, as defined by context information model, by using the *Context Publisher* component. Applications can query and subscribe for the context information using *Context Query and Subscriber*. Context information is provided to those applications by the *Context Provider*. *QoC Evaluator* evaluates the QoC parameters from the information provided in the context object about the QoC sources as described in Section 5. Sensors will be configured from a local configuration file which will have the information about *SourceState*, *SourceType* and components to evaluate the QoC parameters will be configured from a situation configuration file. This situation configuration file will have the information about *LifeTime* and *CriticalValue* of each type of context information concept in context information model for a specific scenario. Our mechanism to evaluate the QoC parameters can be configured according to the requirements of an application or situation by this configuration file. Finally, *Context Annotator* enriches the context information with already evaluated QoC parameters that is provided to subscribed applications by *QoC Provider*.

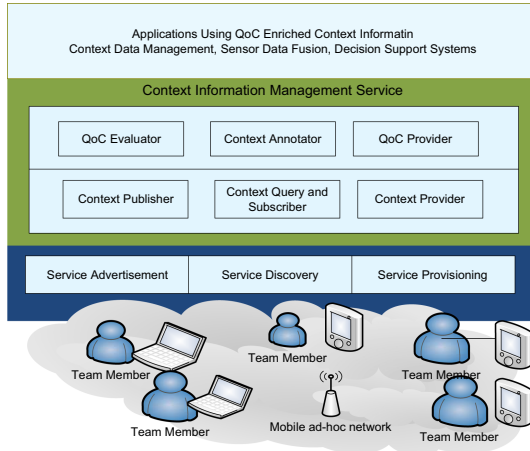


Fig. 4. Components for provisioning of QoC-parameters annotated context information

Sensors, providing information about the machine capabilities, have been implemented using the Intel Mobile SDK [10]. MXQuery [19] and KSOAP2 [18], having a low memory footprint to be able to run on mobile devices, have been used for processing XML data.

8 Conclusion and Future Work

In this work we have discussed the QoC parameters as the worth of context information for a specific application. Selected QoC parameters have also been evaluated and provided to the subscribed applications along with context information. Subsequently, enrichment of context information with Quality of Context parameters by our implemented algorithm enhances the perception of an application about the context information and enables the system to use this context information in an improved way.

For our next steps, we plan to extend this work with the evaluation of the remaining QoC parameters and perform the experiments to evaluate the effectiveness of these parameters. We will also broaden the scope of QoC parameters by discussing the role of reputation and historical information about the source of information and probabilistic nature of the context information in their evaluation. We will examine how the applications using these QoC parameters mention their weights according to their preference, how these parameters depend on each other, and how can we reach a tradeoff between their values. We plan to use QoC parameters in context data management and context information fusion. We will also enhance the quality by combining the context information and QoC parameters represented by more than one context object. We will use this high quality context information to provide robustness in context-aware systems.

References

1. Baldauf, M., Dustdar, S., Rosenberg, F.: A survey on context-aware systems. *Int. J. Ad Hoc Ubiquitous Comput.* 2(4), 263–277 (2007)
2. Bu, Y., Gu, T., Tao, X., Li, J., Chen, S., Lu, J.: Managing quality of context in pervasive computing. In: *QSIC 2006: Proceedings of the Sixth International Conference on Quality Software*, Washington, DC, USA, pp. 193–200. IEEE Computer Society, Los Alamitos (2006)
3. Buchholz, T., Kpper, A., Schiffrers, M.: Quality of context: What it is and why we need it. In: *Proceedings of the 10th International Workshop of the HP OpenView University Association (HPOVUA)*, Hewlett-Packard OpenView University Association (2003)
4. Castro, P., Muntz, R.: Managing context data for smart spaces. *Personal Communications, IEEE [see also IEEE Wireless Communications]* 7(5), 44–46 (2000)
5. Chen, G., Kotz, D.: A survey of context-aware mobile computing research. Technical report, Dartmouth College, Hanover, NH, USA (2000)
6. de Freitas, R., Neto, B., da Graca, M., Pimentel, C.: Toward a domain-independent semantic model for context-aware computing. In: *LA-WEB 2005: Proceedings of the Third Latin American Web Congress*, Washington, DC, USA. IEEE Computer Society, Los Alamitos (2005)
7. Dey, A., Abowd, G., Salber, D.: A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-Computer Interaction* 16, 97–166 (2001)
8. Dey, A., Mankoff, J., Abowd, G., Carter, S.: Distributed mediation of ambiguous context in aware environments. In: *UIST 2002: Proceedings of the 15th annual ACM symposium on User interface software and technology*, pp. 121–130. ACM, New York (2002)
9. Dorn, C., Schall, D., Dustdar, S.: Granular context in collaborative mobile environments. In: Meersman, R., Tari, Z., Herrero, P. (eds.) *OTM 2006 Workshops*. LNCS, vol. 4278, pp. 1904–1913. Springer, Heidelberg (2006)
10. Intel Mobile Platform Software Development Kit 1.3 – Open Source Project for Windows and Linux (August 11, 2008), <http://ossmpsdk.intel.com/>
11. Formayer, H., Frischauf, C.: Extremereignisse und klimawandel in Österreich aus sicht der forschung. Technical report, WWF Austria and Institute of Meteorology, BOKU, Vienna (2004)
12. Gray, P.D., Salber, D.: Modelling and using sensed context information in the design of interactive applications. In: Nigay, L., Little, M.R. (eds.) *EHCI 2001*. LNCS, vol. 2254, pp. 317–336. Springer, Heidelberg (2001)
13. Henriksen, K., Indulska, J.: Modelling and using imperfect context information. In: *PERCOMW 2004: Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, pp. 33–37 (March 2004)
14. Henriksen, K., Indulska, J., Rakotonirainy, A.: Modeling context information in pervasive computing systems. In: Mattern, F., Naghshineh, M. (eds.) *PERVASIVE 2002*. LNCS, vol. 2414, pp. 79–117. Springer, Heidelberg (2002)
15. Honle, N., Kappeler, U.-P., Nicklas, D., Schwarz, T., Grossmann, M.: Benefits of integrating meta data into a context model. In: *PERCOMW 2005: Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications Workshops*, Washington, DC, USA, pp. 25–29. IEEE Computer Society, Los Alamitos (2005)
16. Kim, Y., Lee, K.: A quality measurement method of context information in ubiquitous environments. In: *ICHIT 2006*, pp. 576–581. IEEE Computer Society, Los Alamitos (2006)
17. Krause, M., Hochstatter, I.: Challenges in modeling and using quality of context (qoc). In: Magedanz, T., Karmouch, A., Pierre, S., Venieris, I.S. (eds.) *MATA 2005*. LNCS, vol. 3744, pp. 324–333. Springer, Heidelberg (2005)

18. KSOAP2 (Access Date: August 11, 2008),
<http://sourceforge.net/projects/ksoap2/>
19. MXQuery (Access Date: August 11, 2008),
http://www.dbis.ethz.ch/research/current_projects/MXQuery
20. Ranganathan, A., Al-Muhtadi, J., Campbell, R.H.: Reasoning about uncertain contexts in pervasive computing environments. *Pervasive Computing* 3(2), 62–70 (2004)
21. Razzaque, M.A., Dobson, S., Nixon, P.: Categorization and modeling of quality in context information. In: *Proceedings of the IJCAI 2005 Workshop on AI and Autonomic Communications*, Edinburgh, Scotland (August 2005)
22. Schmidt, A.: A layered model for user context management with controlled aging and imperfection handling. In: Roth-Berghofer, T.R., Schulz, S., Leake, D.B. (eds.) *MRC 2005. LNCS (LNAI)*, vol. 3946, pp. 86–100. Springer, Heidelberg (2006)
23. Sheikh, K., Wegdam, M., van Suinderen, M.: Quality-of-context and its use for protecting privacy in context aware systems. *Journal of Software* 3, 83–93 (2008)
24. Tang, S., Yang, J., Wu, Z.: A context quality model for ubiquitous applications. In: *IFIP International Conference on Network and Parallel Computing Workshops (NPC)*, pp. 282–287 (September 2007)
25. The EU WORKPAD Project, <http://www.workpad-project.eu>
26. Truong, H.-L., Juszczuk, L., Bashir, S., Manzoor, A., Dustdar, S.: Vimoware - a toolkit for mobile web services and collaborative computing. In: *34th EUROMICRO Conference on Software Engineering and Advanced Applications (2008)*
27. Truong, H.L., Juszczuk, L., Manzoor, A., Dustdar, S.: Escape - an adaptive framework for managing and providing context information in emergency situations. In: Kortuem, G., Finney, J., Lea, R., Sundramoorthy, V. (eds.) *EuroSSC 2007. LNCS*, vol. 4793, pp. 207–222. Springer, Heidelberg (2007)
28. van Bunningen, A.H., Feng, L., Apers, P.M.G.: Context for ubiquitous data management. In: *UDM 2005: Proceedings of the International Workshop on Ubiquitous Data Management*, Washington, DC, USA, pp. 17–24. IEEE Computer Society Press, Los Alamitos (2005)
29. Weiser, M.: The computer for the twenty first century. *Scientific American* 265(3), 94–104 (1991)
30. Xu, C., Cheung, S.C.: Inconsistency detection and resolution for context-aware middleware support. In: *ESEC/FSE-13: Proceedings of the 10th European software engineering conference held jointly with 13th ACM SIGSOFT international symposium on Foundations of software engineering*, pp. 336–345. ACM, New York (2005)