Quality-of-Context and its use for Protecting Privacy in Context Aware Systems

Kamran Sheikh
University of Twente, Department of Computer Science, Enschede, Netherlands
Email: k.sheikh@cs.utwente.nl

Maarten Wegdam and Marten van Sinderen
University of Twente, Department of Computer Science, Enschede, Netherlands
Email: {wegdam, sinderen}@cs.utwente.nl

***Abstract*—Context-awareness refers to systems that unobtrusively adapt to the environment of their users on the basis of context information, popularly known as context-aware systems. One inherent property of context information is that it possesses a certain quality, such as the certainty with which it has been determined and so on. Different aspects of this quality are represented by a set of indicators collectively known as Quality of Context (QoC). QoC also represents privacy sensitiveness of context information, i.e. context information of higher quality is considered more privacy sensitive. An important step towards making QoC indicators usable is to quantify them in tangible units. In this paper we provide motivation for using QoC indicators as meta-information for context management and use QoC as part of a user privacy enforcement framework. We propose five QoC indicators and present different alternatives available for expressing them quantitatively.**

***Index Terms*— Context-aware, Quality-of-Context, privacy, user-preferences.**

I. Introduction

A. User context

Context-aware systems present the advantage of offering personalized services to their users based on context information. Dey et al. define context as 'any information that can be used to characterize the situation of an entity', where an entity can be a person, place or object relevant to the current scope of discussion [1]. In ubiquitous systems the most dynamic entities are the human users. We define 'user context' as 'information that describes the situation of a human user either directly or indirectly'. User context needs special consideration for three main reasons. Firstly, user context is the most important subset of context information for personalizing services for users because it describes the situation of these users. This could be direct (e.g. location of a user) or indirect (e.g. the bandwidth available to the user's mobile device). Secondly, human beings can be involved in much more complex and unpredictable situations than any other type of system entity (place or object from Dey's definition). Therefore, sophisticated semantics and methods of determination are required for this type of context. Thirdly, user context is the subset of context information that may disclose users' private information and therefore, privacy protection measures have to be taken at the design stage when this type of context is used [3] [6].

B. Quality of Context (QoC)

Several advantages of having an underlying context management middleware supporting higher level applications with context-specific operations, such as context collection, aggregation and provisioning, have been identified [7][8][21][25][26][27]. One of the main challenges faced in the realization of a context management middleware that can manage heterogeneous context sources and consumers is the 'vagueness' of context information [10]. Context information is characterized by a set of indicators collectively known as Quality of Context (QoC). Buchholz et al. use these indicators to describe how closely a piece of context information reflects the physical reality [2]. We think that quality of context is important for the functioning of a context management middleware for three main reasons [29].

(1) Users' privacy enforcement - The quality of an instance of context information reflects its privacy sensitiveness. Services should not be provided access to context of a higher quality than is needed for the functioning of the services and that too only with explicit user consent. This is stipulated by several privacy legislations such as that of the European Union [24]. Thus, users should be able to express the maximum quality of context that they are willing to share with different requesters to protect their privacy.

(2) QoC-based application adaptation – context-aware applications by definition adapt their behavior based on the user's context. As noted before, context information is inherently imperfect, due to sensor limitations and other reasons. Real-world context-aware applications, therefore, should adapt their behavior to QoC information too. Since this adaptation is highly application dependent, the middleware should pass the QoC along with the

context information to the application, in a consistent manner independent of receiving applications and low-level sensors. In the above tele-monitoring application, the QoC is important when selecting a healthcare giver, e.g., if two health-care givers are approximately equally far away from the patient, the healthcare giver whose status is 'available' with the highest probability is notified (before others).

(3) Middleware efficiency - Gaining access to higher quality context is typically more expensive, in terms of performance, resource usage and cost. If context-aware services specify the needed QoC, the context management middleware can provide context with only the required quality to minimize costs, e.g. by providing a cached value of requested context. We will refer to these three objectives several times in the rest of this article as guiding principles for the choice of our QoC indicators.

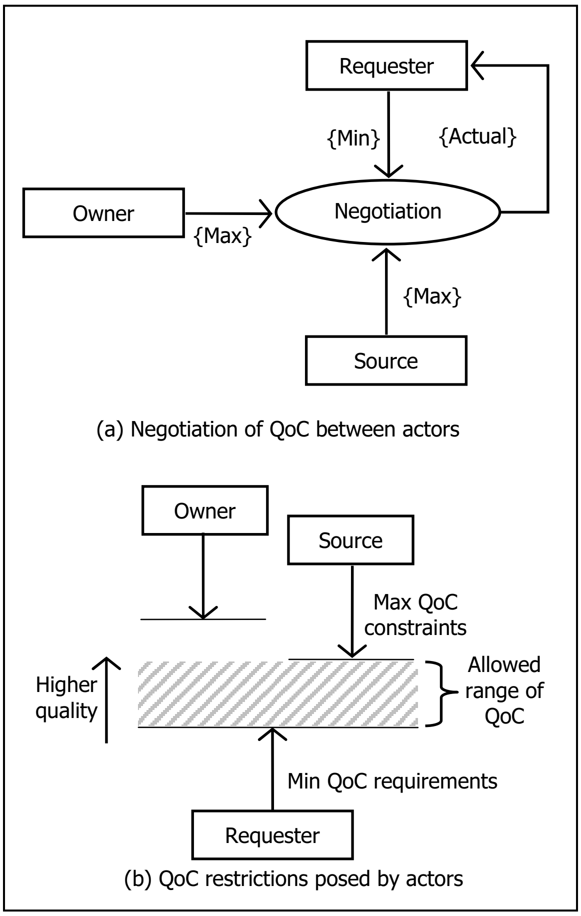

Figure 1. Stakeholders that influence QoC

Reducing the quality of context information to protect user privacy is referred to as 'obfuscation'. Besides obfuscating context information, users may opt to falsify context information itself. Lederer et al. [13] argue that users should not be expected to deviate from normal social practices just because the current technology works differently. Without commenting on its ethical correctness, we would like to point out that falsifying information about oneself is an established practice in social interactions, e.g. screening phone calls. Software applications that disseminate presence related information, such as instant messaging clients, allow their users to set their own presence status enabling them, for instance, to falsely set their status to 'away' while they are at their computer. Context-aware systems that take away this right from their users by disseminating accurate information to others at all times may seriously jeopardize their social acceptability. For this reason, allowing users to falsify context information will allow context-aware systems to make the same provisions for their users.

Three main types of actors interact with a context management middleware and have an influence on the quality of provided context, as shown in Figure 1(a). The context requester communicates a minimum required quality of context to the context management middleware. The context owner, who is the human user whose privacy may be violated by the exchange of the context, enforces an upper limit on the quality of context that is given to the requester through her privacy preferences. The context source is the software entity that provides context to the context management middleware and the maximum quality that it can provide is constrained by its capabilities. For a context interaction to be successfully performed, the limits on QoC set by the context owner (privacy preferences) and the context source (capabilities), both, must be above the QoC requirement of the requester. This situation is depicted in Figure 1(b).

C. *Structure*

The rest of this paper is structured as follows. Section 2 provides an overview of current literature related to QoC. An analysis of our QoC indicators and their application to our three main objectives as presented in Section 3. Section 4 provides as insight into the different options available for the quantification of these QoC indicators which are employed in a real-world scenario explained in section 5. Finally, conclusions and future directions are outlined in section 6.

II. Related work

Buchholz et al. [2] describe five reasons why QoC is necessary as an additional notion of quality. But they are relatively independent of the chosen indicators as the applicability of the indicators to these reasons has not been explained. We have adopted precision, probability of correctness and up-to-dateness (freshness) from their work and explained them in further detail. Trustworthiness has been left out of our current discussion because it does not directly influence our objectives. Furthermore, each context requester has a unique view of how it trusts the other entities it interacts with. This view may be independent of the underlying context management middleware and context sources, e.g. a user might introduce a new trust value for a context sources for reasons unknown to the context management middleware.

Gray and Salber [9] describe six QoC indicators that represent the ambiguity in context information due to sensor inefficiency. The notion of protecting users' privacy through obfuscation of these indicators is not considered. We have introduced some basic improvements such as spatial and temporal resolution (see section III). Also, in our opinion 'repeatability' is one of many ways to measure the 'probability of correctness' of a context source. Therefore we have adopted probability of correctness, which is the broader concept.

Huebscher et. al. [11] present the same indicators as [12] besides trustworthiness, which has been omitted. In their middleware, the 'adaptation engine' can use any number of QoC indicators for activities such as service adaptation and discovery, but the applicability of each of the indicators mentioned has not been described. Ebling, Hunt and Lei [10] describe 'Quality of Information' as a design issue for pervasive systems and mention only two indicators, freshness and confidence. The user's privacy protection point of view has not been considered in both [10] and [11].

Wishart et al. [30] propose a system in which privacy policies are expressed in two distinct levels namely, privacy preference that specifies whether a subject has access to a type of context and granularity preference that specifies the maximum allowable QoC to that subject. But they consider only one QoC indicator, i.e. precision, for this purpose which, in our opinion, is not sufficient for effective privacy protection of end-users. Secondly, they quantify QoC through ontologies while we provide quantification at a lower level. Using our techniques, QoC can be quantified and distributed into the discrete levels as presented in [30].

In general, the objectives for introducing QoC as an additional notion of quality in context-aware systems and the indicators chosen to represent it overlap in existing literature. But all authors fail to highlight the relevance of each of the chosen indicators to their respective objectives and do not provide detail about how each indicator can be quantified so that they can be used for different purposes such as in privacy policies. Finally, Jiang et. al. [19] present a very good technique for enforcing privacy in context-aware systems using 'capturing confidence' (probability of correctness) of sensors and 'representational accuracy' (precision) of the identity of users. But these indicators are insufficient to ensure context management middleware efficiency or express service requirements.

III. Analysis of QoC Indicators

QoC indicators are used by entities that interact with context management middleware to specify constraints on the quality of context (see Figure 1). Based on current literature [2][8][9][10][11] and our experience with context management middleware [14], we have identified five QoC indicators namely, precision, freshness, spatial resolution, temporal resolution and probability of correctness [29]. How each of these indicators is used to fulfill the three objectives presented in section B is described below.

A. Precision

We define precision as the **'granularity with which context information describes a real world situation'**. This QoC indicator describes the granularity with which an instance of context information reflects the real world situation. For example, the information that the temperature of a room is 17.3 degrees Celsius is at a higher precision level than 17 degrees Celsius. Services that utilize context information to provide personalized services to requesters have minimum requirements on the precision of this context. For example, a doctor requesting his patient's body temperature might be interested in a temperature value to the nearest tenth of a degree in degrees Celsius, i.e. with three significant figures. For a weather report however, a temperature value to the nearest degree would be sufficient. A service requesting temperature should, thus, be able to express this requirement to the underlying context management middleware providing the context information. From an context management middleware point of view, when it knows the required precision of the context to be provided, it will be able to utilize context sources that offer the lowest required precision in the hopes of reducing the costs of and making context acquisition more efficient.

From a privacy viewpoint, a user might want to restrict certain requesters from accessing more precise information. Consider a user who is subscribed to a location-aware weather service that provides the weather according to the city in which she is currently present. Her location is collected from her GPS device at the precision level of +/- 10m, but she wants to 'obfuscate' its precision to reveal only the name of the city to the weather service.

B. Freshness

We define freshness as **'the time that elapses between the determination of context information and its delivery to a requester'**. Figure 6 gives a pictorial depiction of this concept. Freshness plays a very important role for all of the three objectives mentioned in Section I. As an example of service requirement, a patient's body temperature older than 2 hours may be too old for a requesting doctor. For any context management middleware, one of the most expensive operations is to collect context information from the environment in real-time. Caching information at a proxy site may make collection of context information more efficient but has a direct impact on its freshness, as shown in Figure 2. If the context management middleware is aware of the freshness required by client applications, it could employ caching mechanisms efficiently, which would reduce the cost of context provisioning. Freshness can also be used to protect users' privacy, for example, a user may allow others to access where she has been 3 days earlier, but not any newer information.

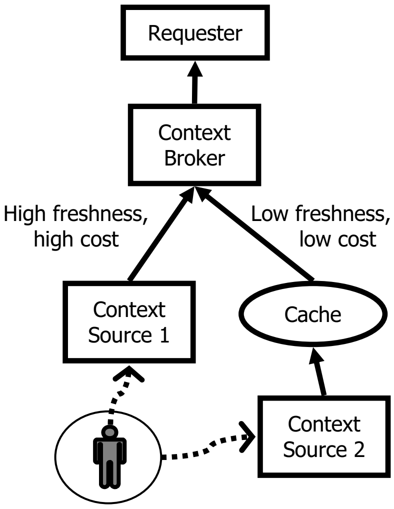

Figure 2. Freshness and cost

C. *Spatial Resolution*

We define spatial resolution as **'the precision with which the physical area, to which an instance of context information is applicable, is expressed'**. Thus, spatial Resolution refers to the area in physical space to which some context information is associated to. When context information describes one or more aspects of physical space, it is often implied that it is a reasonable estimate with regards to spatial resolution. For example, when we say that the temperature in Enschede city is 25°C, it is not true for the whole city. In the same way, when we quote the temperature of a room, we do not take into account the lower temperature near the window or the higher temperature near the heating radiators. Figure 3 gives a pictorial representation of this concept.

When a service requests the temperature from a context source, it has to specify the area which it is interested in, e.g. a room in a hospital, a building or the city. The context management middleware would be able to select more cost-effective context sources if the minimum spatial resolution required by requesters is known, e.g. getting the temperature of a specific location would be more expensive than an average value for the city. Users might want to reduce spatial resolution of context information for privacy reasons. For instance, users may allow a building security system access to the number of people in their building but not in the room in which they are present. This will prevent the system from deducing whether the user is in a meeting. Clearly, the property of spatial resolution is relevant only to context information that is about physical space such as temperature or number of people present. Information such as 'Is John at work?' does not have a spatial resolution.

D. *Temporal Resolution*

We define temporal resolution as **'the period of time to which a single instance of context information is applicable'**. Like space, a context also has a breadth of time to which it is applicable. Temporal resolution shows the best possible approximation of time at which a context was determined. Figure 4 illustrates this by showing two samples collected by a sensor. The period of time between the collected samples is a limitation on the granularity of the time of determination which signifies temporal resolution.

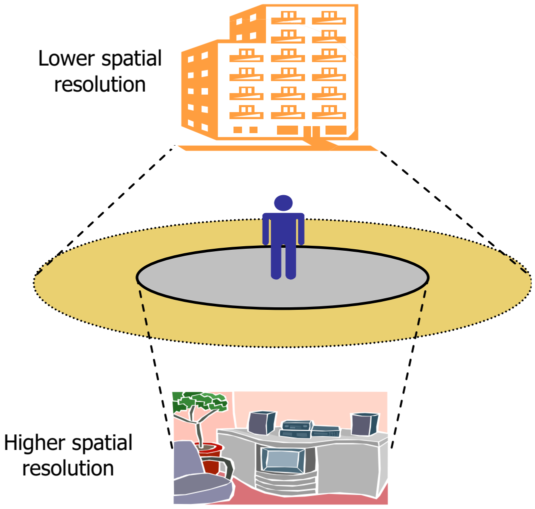

Figure 3. Spatial resolution

Consider, for example, the temperature of a room that is collected by one or more sensors every 8 hours. So, one collected value of this context is applicable for 8 hours which is its maximum temporal resolution. This is an important quality requirement that services need to communicate to the context management middleware, e.g. a doctor requires a patient's room temperature measured every 2 hours. The context management middleware can select context sources with the least required temporal resolution if this requirement is known and may even instruct context sources (e.g. temperature sensors) to collect values with optimal frequency to reduce costs. Temporal resolution plays a vital role in protecting user privacy too. For example, instead of seeing that Bob left the office at 29-06-2006 1632hrs, a requester might only be told that he left on 29-06-2006. The temporal resolution, in this case, has been dropped from the nearest minute to the nearest day (see Figure 4).

Temporal resolution is different from freshness that is explained earlier. While freshness refers to how old a piece of context information is, temporal resolution is the duration during which the context information might have been true in physical space. To show the difference between the two, in the above example, information about Bob leaving the office might be from 29-07-2006 instead of 29-06-2006 which shows that even though it is fresher (one month apart), it has the same temporal resolution (see Figure 6).

A separation needs to be made depending on whether the context is an event or a state of the user. A state remains true for a continuous period of time and possesses a 'time of determination'. This is the time when the state was determined to be true without any indication about when the state started and when it would end (or has ended). For example, "Is Bob in his office?"

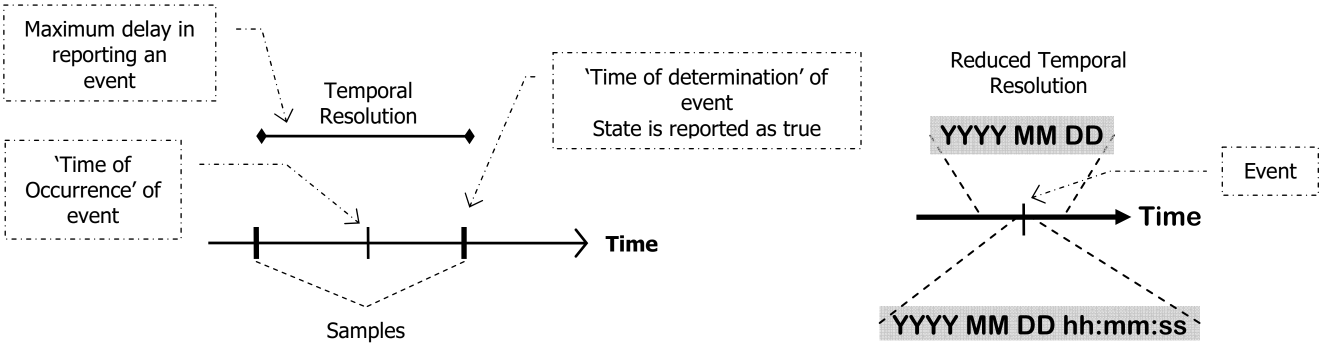
Maximum delay in reporting an event
'Time of Occurrence' of event
Temporal Resolution
'Time of determination' of event
State is reported as true
Time
Samples
Reduced Temporal Resolution
YYYY MM DD
Event
Time
YYYY MM DD hh:mm:ss

Figure 4. Temporal Resolution

An event is a change in state and theoretically does not occupy a period of time. It has the same start and end time. Therefore it possesses a 'time of occurrence' and a 'time of determination'. The time of determination may be at any time later than the time of occurrence, e.g. 'Bob walked in through the front door. Time of occurrence = 1025hrs, time of determination=1050hrs'. Figure 5 shows a UML representation of these concepts.

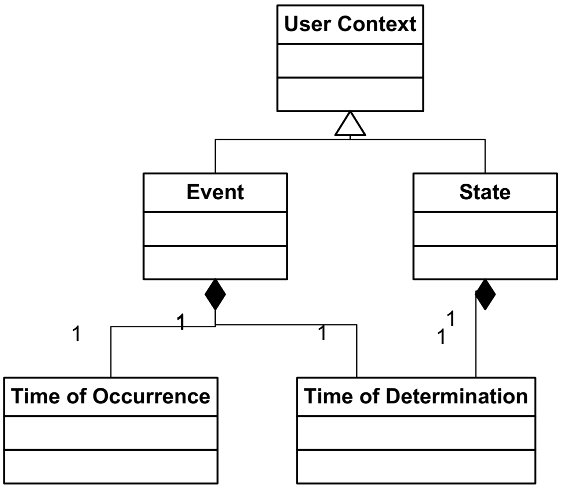

Figure 5. UML Representation of User Context

E. Probability of correctness

We define probability of correctness as **'the probability that an instance of context accurately represents the corresponding real world situation, as assessed by the context source, at the time it was determined'**. There are several reasons due to which the context information being provided may be unintentionally incorrect. Probability of correctness (PoC) refers to the confidence of the source that the provided context information was accurate at the moment it was determined. Consider, for example, a system in which the temperature change, when the heating system in a room is switched on, is used to sense someone's presence there. This sensed context would have a low probability of correctness as the room takes time to heat up and the user might have left the room leaving the heating switched on. One way to increase PoC of context is to employ redundant sources of the same information and check them for consistency. As shown in Figure 7, this has an obvious impact on cost of context acquisition.

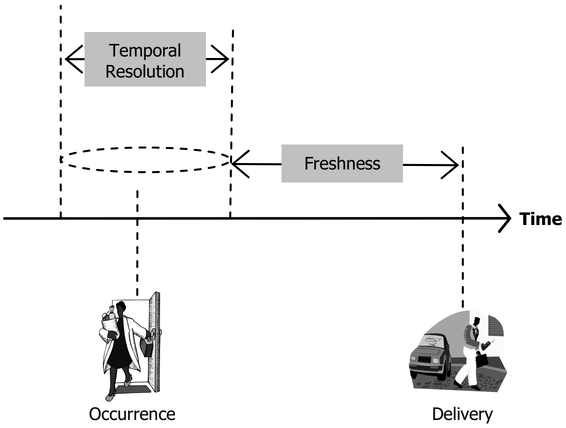

Figure 6. Freshness and temporal resolution

Different applications require context information at different levels of probability of correctness, for example, a security service that opens the building gates when an employee is standing outside the door would need the employee's location with very high probability of correctness. On the other hand, a location-aware weather service can work with a lower probability of correctness because the maximum harm with a wrong location would be that the user is provided wrong weather. With the user's privacy protection point of view, probability of correctness of context information will play its part for 'plausible deniability' [19]. Users may be obliged to allow certain requesters access to their location but they can 'artificially' reduce its probability of correctness so that they can later deny being at certain locations.

PoC serves as an umbrella for several other concepts such as the <method> element that is part of the GEOPRIV Location object [22] so that the receiver can estimate the correctness of the information. But this is privacy sensitive information, especially when the location is deliberately obfuscated. The 'repeatability' indicator in [9] also serves a similar purpose. These pieces of information can be used to compute a coarse-grained PoC value which would help protect users' privacy better and be independent of the method employed to determine its value.

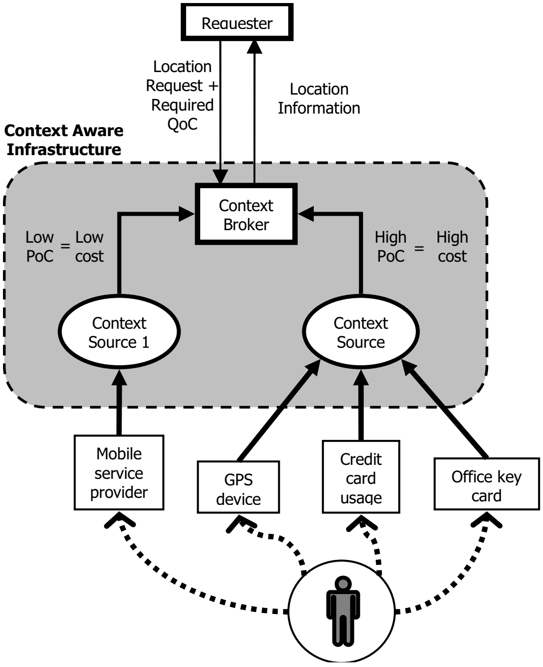

Figure 7. Increasing PoC using redundant context sources

IV. Quantification of QoC Indicators

We already explained the three major reasons why we think quality of context is required in context management middleware and how each of the proposed five QoC indicators are used to fulfill these purposes. Quantifying QoC indicators is the first step towards making QoC usable in the different functional aspects of a context management middleware. How each of them is interpreted, quantified and represented is highly dependent upon the type of context information and the specific application in question.

Context information always possesses a certain quality, whether or not it is explicitly stated, e.g. a temperature value of 25.3°C has a precision of one-tenth of a degree which is true even if not articulated. Figure 1(a) shows four situations in which QoC is communicated from one actor to the other. QoC will have to be stated clearly when (1) context sources advertise their capabilities, (2) context owners specify the maximum allowable QoC and (3) context requesters express their minimum requirement. This is because in these situations QoC requirements are being expressed and no actual context gets exchanged. When context information is provided to a requester, QoC information may be communicated implicitly with other information, e.g. from a timestamp, the freshness and temporal resolution associated with context may be inferred. Nonetheless, values of other QoC indicators have to be explicitly stated, e.g. a requester can not infer the probability of correctness or spatial resolution of some context that it has received.

Following is a discussion about different alternatives available for the quantification of QoC indicators.

A. Precision

For the purpose quantifying precision, context information can be divided into four different types.

1) Boolean

These are context types that can have only a true or a false value. This type of context information can not possess different levels of precision. The uncertainty with which a certain boolean context is known is represented by the probability of correctness discussed later in this section.

2) Numeric

This is the type of context information that can be completely represented by one or more numerical value(s), e.g. speed, distance, temperature. Precision of these can be quantified in two different ways.

- *Ranges:* A range of values within which the real value lies may be specified. These ranges usually correspond to the purpose for which the context information is being asked, e.g. for the purpose of calculating correct fare in a public transport vehicle, the system might only be told whether the subject is within 1-5km, 5-15km or more than 15km, instead of providing the exact known distance from current location to destination.
- *Significant figures:* The precision of a measurement represented by a numeric value is depicted by the number of significant figures in it, e.g. the distance measurement 1.63m has 3 significant figures while 1.6m has 2. Reducing the number of significant figures in such a value is a trivial task and can be used effectively to obfuscate such context information. In effect, significant figures imply course grained ranges, for example, a measurement of 1.6m implies that the real value is between 1.55 and 1.65m.

3) Complex types with an incremental structure

For a context type that can not be represented by numeric values, the first step would be to identify whether its composite parts can be arranged into an ordered series of sets that represent increasing information value. For example, IETF RFC 4119 [22] provides standard notations to express the various components of a civic location. A European civic location can be completely stated using 4 of these namely, Country, A3 (city), A6-STS (street), HNO-HNS (House number with suffix). These can be arranged into a series which progressively represent higher information value.

- (Country)
- (Country, A3)
- (Country, A3, A6-STS)
- (Country, A3, A6-STS, HNO-HNS)

As is evident from the above discussion, obfuscating such types of context information is not trivial. A generic methodology to quantify precision is not possible because it is different for each context type. The number of levels at which the precision of such context types can be expressed has to be fixed and agreed upon by all parties involved, in a prior setup phase.

4) Unordered Complex Types

Finally, there are types of context information that do not fall in any of the above categories. For example, the mood of a user is a context type for which a precision can not be defined. In IETF RFC 4480 [28] a list of mood names has been provided that can be used by applications.

B. Spatial Resolution

Quantification of this QoC indicator is very highly dependent on the type of context information being requested and the specific application requesting the context information. Below, we provide two examples.

- When an application requests the current activity of a user, the spatial resolution may be expressed as a radius around a GPS coordinate. Different levels of spatial resolution may be explicitly stated using different radii. In this example, spatial resolution is stated explicitly by the radius.
- A building security system that keeps track of the number of people present in the building may provide this information with the spatial resolution of a room, a floor, a section of the building or the whole building. A number of location expressive models are available that can be used for this purpose. For example, using the civic location elements specified in IETF RFC 4119 [22] , the following three levels can be defined for a building. Spatial resolution in this way may be stated explicitly by choosing one of these levels or may be implicit in the location information itself.
  - LMK (building name; lowest spatial resolution)
  - LMK, FLR (building name + floor)
  - LMK, FLR, LOC (building name + floor + room; highest spatial resolution)

The Open Geography Markup Language [23] can be used to formally express more complex geographical topologies.

C. Temporal Resolution

Every piece of context information is associated to a point in time when it was collected or when it actually occurred. This information is crucial for its completeness as context information without any indication about the duration of time it is associated to, is not much use. Temporal resolution can be considered the precision of the time of occurrence/determination of context information. Therefore, how it gets quantified depends on how the time of measurement is expressed. Below we provide three examples.

- Unix timestamp: A Unix timestamp is the number of nano-/milli- seconds that have elapsed from 01-Jan-1970 0000hrs (called the 'epoch') to the point when the event occurred. This can be considered as a numerical value and its precision can be quantified as explained earlier (see Section II).
- Standard date/time: constitutes of year, month, day, hours, minutes and seconds. These can be arranged as a series of sets that represent increasing information value (see Section III). Notably, in the standard date/time and Unix timestamp examples, temporal resolution may be expressed implicitly by the timestamp.
- Time period: Instead of depending on the particular format of the timestamp, the context owner could just specify a minimum period of time in the range of which the time of occurrence/determination can be expressed. For example, when a party requests an event that occurred at 1530hrs and the context owner has specified a maximum temporal resolution of 8 hrs, the requester could be told that the event occurred between 1400hrs and 2200hrs. The bounds of this time period can be random as long as the minimum length is 8 hrs and the actual time of occurrence of the event falls within the range. Alternatively, a day/week/month/year can be divided into two or more parts and the requester can be told what subsection of time the event belongs to, e.g. 0001-0600hrs, 0601-1200hrs, 1201-1800hrs and 1801-0000hrs. In this way, the temporal resolution of context information is expressed explicitly.

D. Freshness

Services can express their requirement of minimum freshness (i.e. maximum age) and users can state a minimum age (i.e. maximum freshness) of provisioned context using any suitable unit of time. For example, a service can constrain incoming user context to be no more that 1 hour old, while a user may restrict context information any newer that 30 minutes to be provided to the service. Then, context information that is between 30 and 60 minutes may be provided to the service.

E. Probability of correctness

Context sources can express their level of confidence in an instance of context information in several ways. Some examples are,

- As a percentage value between 0% and 100%.
- A more course grained approach would be to define levels such as low, medium and high.

Notably, probability of correctness is often inversely proportional to other QoC indicators including precision, spatial and temporal resolution. For example, the probability-of-correctness of context information that an employee entered the office would be much higher at a temporal resolution of hours (e.g. between 8-9AM) than seconds (e.g. 08:36:19AM) due to limitations in the capabilities of sensors. In the same way a user's location at country precision can be determined with higher probability of correctness than at street level by a mobile service provider.

V. QoC-Aware Privacy Policy Framework

To demonstrate how user privacy can be protected using the proposed QoC indicators, we describe a real-world scenario where we will use the quantified QoC parameters described above to create privacy policies through which users can specify the QoC that may to be provided to requesters in different situations. Telemonitoring is a process in which different physiological variables of patients, who require constant medical supervision, are measured and assessed while the patient is not present in a hospital. One of the implementation scenarios of the context management middleware proposed by the AWARNESS project [14] is that of an epileptic patient [15]. A patient experiencing an epileptic seizure may lose all control of herself and start shaking rapidly. She immediately needs attention of another person who can move her away from dangerous objects and provide immediate care. In the AWARENESS telemonitoring scenario, patients wear a set of sensors that collect real-time health information and forward to a PDA for processing. These devices are collectively known as the Body Area Network (BAN) [16]. The health signals collected get assessed in the PDA and may be communicated to another location for more processing or viewing by a health care professional. Different available networks may be used for this purpose as shown in Figure 8. The scenario is discussed with more detail in [17].

The AWARENESS context management middleware, in this scenario, is used to collect context information about two main types of users namely, patients and caregivers. Caregivers include medical doctors and other people who have volunteered to help the patient in an emergency, such as family members, friends and colleagues. Table 1 shows the types of context that can be collected and the corresponding maximum quality that may be provided to the health monitoring service. Thus, this set of privacy policy rules would be used to prevent the health monitoring service from receiving any privacy sensitive information about users and caregivers without an emergency.

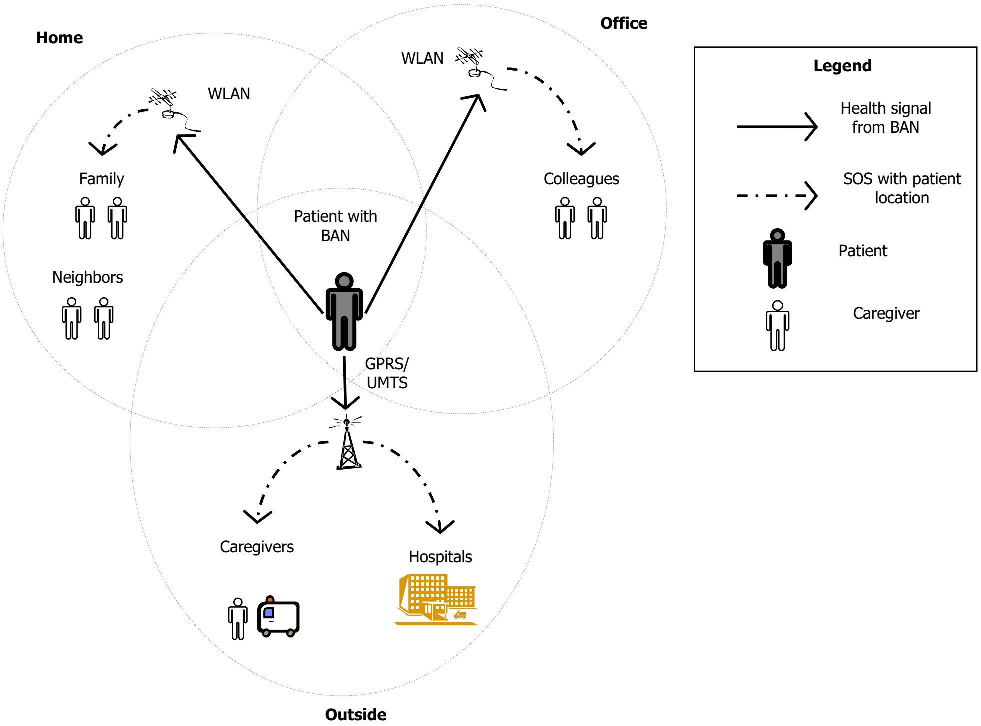

Figure 8. Epileptic seizure patient in different situations

In the interest of brevity, we will demonstrate privacy policies made for only one context type, i.e. 'caregivers in vicinity' which represents the information about caregivers who are in the area surrounding the patient in the GeoPriv Common Policy format [18]. The format shown in Table 1 can be used to specify minimum quality requirements (expressed as a number) by a requester, as well as advertisement of maximum QoC provided by context sources.

Table 1. High-level patient privacy policy for telemonitoring

	Non-emergency	Emergency
Location of patient	(Country, City)	(Country, city, postcode, street, house number)
caregivers in vicinity	Number of available caregivers in a 50km radius from patient location	Identities, locations and estimated-times-of-arrival (ETAs) of available caregivers in a 5km radius from patient location
Available bandwidth	Within a ranges 0-50, 50-200, above 200 kbps	Exactly available
Calendar appointments	No access	Access to all appointment details within +/- 2hrs of time of emergency.

Table 2. QoC-based privacy policy for context type 'caregivers in vicinity'

	Precision	Spatial Resolution	Temporal Resolution	Freshness	Probability of Correctness
Non-emergency	1	3	1	3	2
Emergency	3	2	3	3	3

Table 3. QoC levels for 'caregivers in vicinity' available at Context Registry

	Precision	Spatial Resolution	Temporal Resolution	Freshness	Probability of Correctness
0	No access	No access	No access	No access	No access
1	Number of available caregivers	Within 20 meters of patient	(Year, month, day)	Older than 24 hours	Low
2	Identities of available caregivers	Within 5km of patient	(Year, month, day, hour)	Older than 2 hours	Medium
3	IDs and locations of available caregivers	Within 50km of patient	(Year, month, day, hour, minute)	Freshest available	High

Figure 9 illustrates the software architecture of the QoC aware privacy enforcement of the AWARENESS context management middleware. The context registry contains a database that stores all the available context types and the corresponding level of QoC offered by all registered context sources. Table 3 shows a sample format of how the context registry can store the different levels of quality of user context that various context sources can provide. The 'Users Privacy Preferences' database contains policies of the form shown in Table 2. The context registry, in addition to storing context source capabilities, also stores tables like Table 3 which map a {context type, QoC indicator, quality level) tuple to a set of obfuscation directives for the context processor. The context broker provides these obfuscation directives to the context processor which acts as a gatekeeper for the health monitoring service.

The context broker uses the context registry database to map a set of {context type, QoC indicator, quality level} tuples to a set of context sources that can offer the context information at that level. The details of such an architecture is out of scope of this document.

VI. Conclusions and Future Work

In this paper we have explained the significance of Quality of Context information for context management middleware and the contribution towards its negotiation by three main stakeholders interacting with such a context management middleware. Five QoC indicators have been proposed and different options available for their quantification have been discussed. Finally, the applicability of the QoC indicators and schemes for their quantitative expression are demonstrated through a health tele-monitoring scenario in which the privacy of users is protected through a QoC-based privacy policy framework.

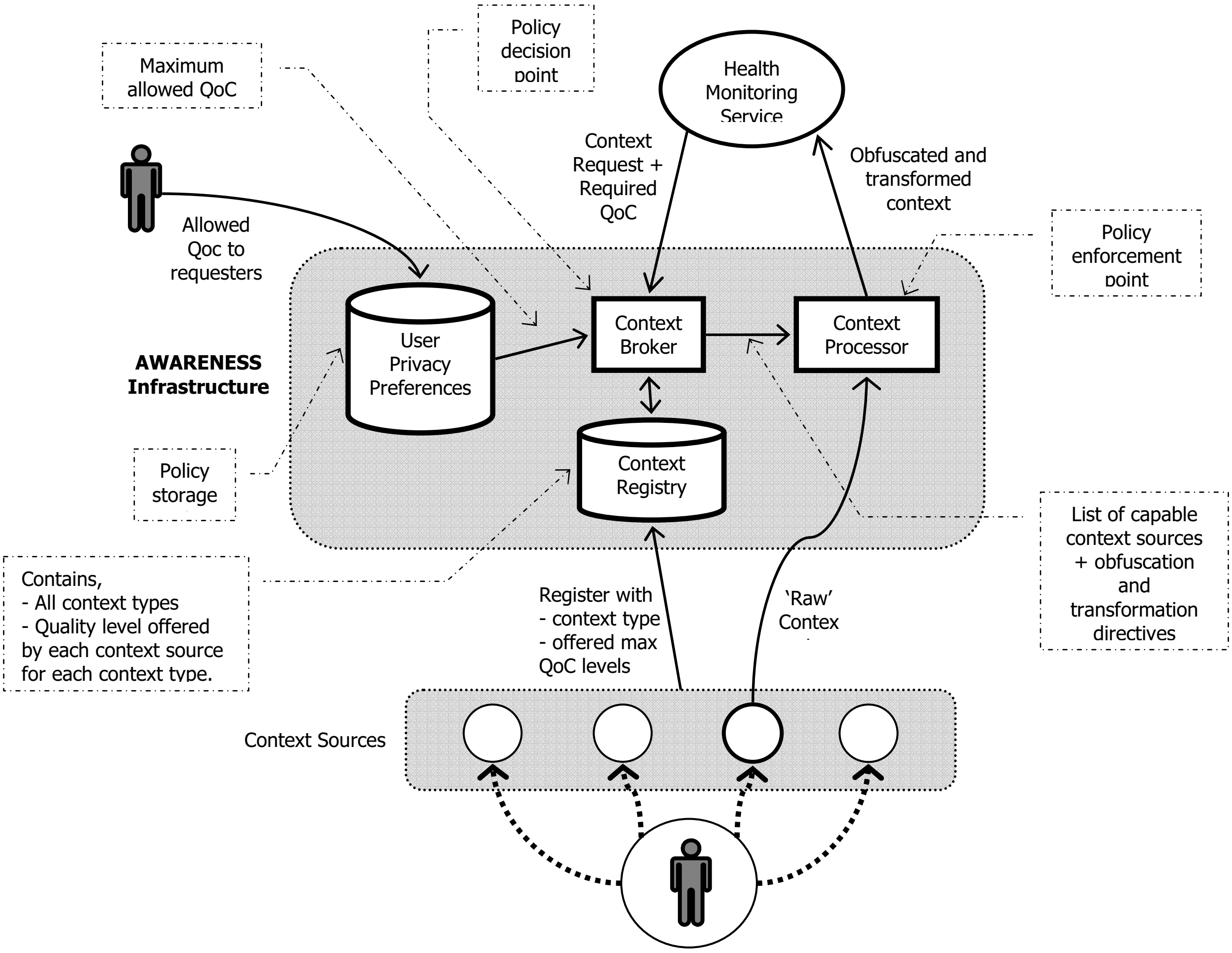

Figure 9. Middleware Architecture for Privacy Protection

In the future we plan to study the feasibility of a sixth QoC indicator named 'trustworthiness' [2] which we omitted from our current discussion because of its inherent complexity. Unlike the other indicators, each entity in a context-aware system has a unique view of how it trusts other entities which makes trust evaluation very complex for the context-aware middleware. The techniques for quantification of the QoC indicators presented in this paper will be used to express (1) Application requirements (2) Context source capabilities and (3) Users' privacy preferences. Then a policy standard, such as the GeoPriv Common Policy format [18], will be used at runtime to negotiate the final level of QoC that is communicated to requesters.

Acknowledgements

This work is part of the Freeband AWARENESS project (http://awareness.freeband.nl). Freeband is sponsored by the Dutch government under contract BSIK 03025.

This work is also part of the Amigo project, funded by the European Commission as an integrated project (IP) in the Sixth Framework Programme under contract number IST 004182.

References

[1] A.K Dey, "Understanding and Using Context," J. Personal and Ubiquitous Computing, vol. 5, no. 1, Feb. 2001, pp. 4-7.

[2] T. Buchholz, A. Küpper, and M. Schiffers. Quality of context: What it is and why we need it. In Proceedings of the Workshop of the HP OpenView University Association 2003 (HPOVUA 2003), Geneva, 2003.

[3] Lederer S, Mankoff J, Dey AK, Beckmann C (2003) Managing personal information disclosure in ubiquitous computing environments. Technical report CSD-03-1257. University of California, Berkeley, California

[4] F. B. Schneider. Enforceable security policies. ACM Transactions on Information and System Security, 3(1):30-50, February 2000.

[5] P. Coy. Big Brother Pinned to your Chest. Business Week, 3279, August 17, 1992.

[6] Langheinrich, M.: Privacy Invasions in Ubiquitous Computing. Workshop on Socially-informed Design of Privacy-enhancing Solutions in Ubiquitous Computing. UbiComp 2002, Göteborg, Sweden.

[7] Dey, A. K., Salber, D., & Abowd, G. D. (1999). A context-based infrastructure for smart environments. Proceedings of the 1st InternationalWorkshop on Managing Interactions in Smart Environments (MANSE 99). Heidelberg, Germany: Springer-Verlag.

[8] Jason I. Hong and James A. Landay. An infrastructure approach to context-aware computing. Human-Computer Interaction, 16(287-303), 2001.

[9] Phil Gray and Daniel Salber. Modelling and Using Sensed Context Information in the design of Interactive Applications. In Proceedings of the 8th IFIP Working Conference on Engineering for Human-Computer Interaction (EHCI 01), Toronto, Canada, May 2001.

[10] Ebling, M., Hunt, G.D.H., Lei, H.: Issues for context services for pervasive computing. In: Middleware 2001 Workshop on Middleware for Mobile Computing, Heidelberg (2001)

[11] M. C. Huebscher and J. A. McCann. Adaptive middleware for context-aware applications in smart-homes. In Proceedings of the 2nd Workshop on Middleware for Pervasive and Ad-Hoc Computing (MPAC), Oct. 2004.

[12] Hung Q. Ngo et al: Developing Context-Aware Ubiquitous Computing Systems with a Unified Middleware Framework. In Proceedings of the EUC 2004: 672-681.

[13] Lederer, S., J.I. Hong, A. Dey, and J.A. Landay, 2004. Personal Privacy through Understanding and Action: Five Pitfalls for Designers. Personal and Ubiquitous Computing 8, 6, 440 - 454.

[14] M.J. van Sinderen, A.T. van Halteren, M. Wegdam, H.B. Meeuwissen, E.H. Eertink, Supporting Context-aware Mobile Applications: an Infrastructure Approach, IEEE Communication Magazine, September 2006.

[15] AWARENESS project results, Deliverable 1.1v2; http://awareness.freeband.nl

[16] Halteren,. A., Bults, R., Widya, I., Jones, V., Konstantas, D., Mobihealth-Wireless Body Area Networks for Healthcare, Proc. New generation of wearable systems for e-health 2003, pp121-126, 2003

[17] T. Broens, A. van Halteren, M. van Sinderen, K. Wac, Towards an application framework for context-aware m-health applications, Proceedings of 11th Open European Summer School (EUNICE 2005), 6-8 July 2005, Colmenarejo, Spain

[18] IETF Internet Draft, Common Policy: A Document Format for Expressing Privacy Preferences, http://tools.ietf.org/wg/geopriv/draft-ietf-geopriv-common-policy/ (accessed 17 Aug 2006)

[19] X. Jiang and J. A. Landay, "Modeling Privacy Control in Context-Aware Systems", in 1(3), pp. 59-63, IEEE Pervasive Computing, 2002.

[20] R. Hull, B. Kumar, D. Lieuwen, P. Patel-Schneider, A. Sahuguet, S. Varadarajan, and A. Vyas. Enabling Context-Aware and Privacy-Conscius User Data Sharing. In Proc. of the International Conference on Mobile Data Management, pages 187-198, IEEE, 2004.

[21] Salber, D., Dey, A.K., and Abowd, G.D. The Context Toolkit: Aiding the development of context enabled applications. In Proceedings of CHI'99 (1999) 434-441.

[22] IETF RFC (4119). A Presence-based GEOPRIV Location Object Format. http://www.ietf.org/rfc/rfc4119.txt?number=4119

[23] OpenGIS, "Open Geography Markup Language (GML) Implementation Specification", OGC 02-023r4, January 2003, <http://www.opengeospatial.org/specs/?page=specs>.

[24] Michael Friedewald, David Wright, Elenena Vildjiounaite (ed.). Safeguards in a World of Ambient Intelligence (SWAMI): Scenario Analysis and Legal Framework - First Results. Report submitted to the participants of the first SWAMI expert workshop, held in Brussels, 1 June 2005.

[25] Henricksen, K., J. Indulska, et al. (2005). Middleware for Distributed Context-Aware Systems. On the Move to Meaningful Internet Systems 2005, Agia Napa, Cyprus, Springer Berlin / Heidelberg.

[26] Gu T., Punga H. K., Zhang D. Q. A service-oriented middleware for building context-aware services. Journal of Network and Computer Applications archive, Volume 28, Issue 1 (January 2005)

[27] Chan A. T. S., Chuang S. N. MobiPADS: A Reflective Middleware for Context-Aware Mobile Computing. IEEE Transactions on Software Engineering, vol. 29, no. 12 (December 2003)

[28] IETF RFC (4480). RPID: Rich Presence Extensions to the Presence Information Data Format (PIDF). http://www.ietf.org/rfc/rfc4480.txt?number=4480

[29] K.Sheikh, M.Wegdam, M.Sinderen. Middleware Support for Quality of Context in Pervasive Context-Aware Systems. In proceedings of Perware 2007 workshop held in conjunction with the 5th Conference on Pervsive Computing and Communications (Percom 2007), White Plains, NY, USA, Mar 2007.

[30] Wishart, R., Henricksen, K., Indulska, J.: Context obfuscation for privacy via ontological descriptions. In: 1st International Workshop on Location and Context Awareness. Volume 1678 of Lecture Notes in Computer Science, Springer (2005) 276-288

Kamran Sheikh (k.sheikh@cs.utwente.nl) holds a Master's Degree in engineering science majoring in Computer Science from the University of New South Wales, Australia and is currently a PhD student at the University of Twente in The Netherlands. His research interests include, context-aware middleware and end-user privacy enforcement. He is currently involved in two major research projects. Firstly, the Dutch Freeband AWARENESS project (BSIK 03025) which aims to build an infrastructure to support context-aware mobile applications. Secondly, the Amigo project, funded by the European Commission as an integrated project (IP) in the Sixth Framework Programme under contract number IST 004182.

MARTEN J. VAN SINDEREN (sinderen@cs.utwente.nl) received his Master's degree in electrical engineering and Ph.D. degree in computer science from the University of Twente, The Netherlands. He is an associate professor at the University of Twente, and manager of A-Services Internet, one of the strategic research orientations of the Centre for Telematics and Information Technology, the ICT research institute of the University of Twente. His research interests include design methods and architectures for networked systems, and service platforms for supporting context- aware mobile applications. He currently leads the Dutch Freeband A-MUSE project on service design and semantic interoperability.

MAARTEN WEGDAM (wegdam@cs.utwente.nl) holds a Master's and a Ph.D. degree in computer science from the University of Groningen, The Netherlands, and the University of Twente, respectively. He is a senior member of technical staff of Bell Labs Europe at Alcatel-Lucent in The Netherlands. His fields of interest include context awareness, identity federation, privacy, middleware, and QoS in middleware. He currently leads the Dutch Freeband AWARENESS project on an infrastructure to support context-aware mobile applications. He has a parttime position as assistant professor at the University of Twente.