A Multi-Layer Framework for Quality of Context in Ubiquitous Context-Aware Systems

Asma Abdulghani Al- Shargabi
Faculty of Computing and Information Technology
University of Science and Technology
Sana'a, Yemen
a.alshargabi@gmail.com

Francois Siewe
Software Technology Research Laboratory (STRL)
Faculty of Technology
De Montfort University
Leicester, UK
fsiewe@dmu.ac.uk

Abstract-This paper proposes a novel framework for Quality of Context (QoC) in context-aware systems. The main innovative features include: (1) a new definition that generalizes the notion of QoC; (2) a novel multilayer context model; (3) a novel model of QoC that introduces new quality parameters; (4) a novel mechanism to define QoC policy by assigning weights to QoC parameters using a multi-criteria decision-making technique; (5) and a novel quality control algorithm that handles context conflicts, context missing values, and context erroneous values. Our frameworkis implemented in MatLab and evaluated using a case study of a flood forecast system.

Keywords— Context-aware Systems; Quality of Context; Quality Control; Quality Parameters; Quality Policy.

I. INTRODUCTION

The term ‘Ubiquitous’, which means appearing or existing everywhere, was combined with the term "Computing" to form the term "Ubiquitous Computing", which is used to describe ICT (Information and Communication Technology) systems that enable information and tasks to be available everywhere. Devices should vanish into the background to make the user and his tasks the central focus rather than computing devices and technical issues [36][37]**.** This vision has seen a remarkable development where the physical world environment is being increasingly digitally instrumented and strewn with embedded sensor-based and control devices [36]. Context-Aware computing is a field in the wider ubiquitous computing. Context-aware systems use context information to decide what adaptation actions to perform in response to changes in their environment. Depending on applications, context information includes physical context (e.g. temperature and location), user context (e.g. user preferences and user activity), and ICT context (e.g. device capabilities and battery power). Sensors are the main means of capturing context. Unfortunately, sensed context data are commonly prone to imperfection due to the technical limitations of sensors, their availability, dysfunction, and the highly dynamic nature of environment. Consequently, sensed context data might be imprecise, erroneous, conflicting, or simply missing. This imperfection affect the quality of context in higher level of context (i.e. derived context). To manage the impact of context imperfection on the behavior of a context-aware system, a notion of Quality of Context (QoC) is used to measure the quality of any information that is used as context information. Adaptation is performed only if the context data used in the decision-making has an appropriate quality level.

This paper proposes a novel framework for QoC in context-aware systems, which is called MCFQoC (Multilayered-Context Framework for Quality of Context). The main innovative features of the framework include: (1) a new definition that generalizes the notion of QoC to encompass sensed context as well as user profiled context; (2) a novel multilayer context model that distinguishes between three context abstractions: context situation, context object, and context element; (3) a novel model of QoC which extends the existing models with new quality parameters and explicitly distributes the quality parameters across the three layers of context abstraction; (4) a novel mechanism to define QoC policy by assigning weights to QoC parameters using a multi-criteria decision-making technique called Analytical Hierarchy Process (AHP); (5) and finally, a novel quality control algorithm called IPQP (Integrating Prediction with Quality of context Parameters for Context Quality Control) for handling context conflicts, context missing values, and context erroneous values. Our framework, MCFQoC,has been implemented in MatLab and evaluated using a case study of a flood forecast system. Results show that the framework is expressive and modular, thanks to the multilayer context model and to the notion QoC policy which specifies the importance of individual context parameters.

The rest of the paper is organized as follows. The related work is presented in Section II. The proposed framework is described in Section III. The Section IV talks about the implementation of the

framework, while Section V presents the evaluation of the proposed framework. Section VI concludes the paper.

II. Related Work

For context-aware systems, context inherently is imperfect. Conflicts, erroneous, imprecision, and even unknown and missing values could be found within the context information values [5]. These context imperfect aspects led us to think about a solution that contributes in raising the quality level of context. In addition, when we were investigating the QoC in the literature, we have remarked many points that have inspired us to think about developing a comprehensive framework for QoC to figure out the context imperfection aspects, and offer an effective method for context quality control. The points that have been remarked in the literature related to QoC can be summarized as follows:(1) Most work achieved in this area deals with context quality issues separately and not as integrated parts. There are many issues related to QoC that have been addressed in the literature such as context quality parameters [1][2][3][7][8][11][30][31][47], context quality measures [2][7][30], resolving context conflicts [9][10][15][16], context validity, and resolving context uncertainty [30][32][33][34]. However, realizing quality aspects needs to a comprehensive solution that can integrates all these issues together.

(2) Many solutions, which are introduced in the literature for context quality, have been built depending on a simple view of the context, whereas the context has a complex taxonomy and different views. Thinking about QoC should start by thinking about the nature of the context itself. For example, context information combines different levels of context. In the low level, there are context elements or context facts, which are aggregated to compose a higher context information (abstracted context), and then abstracted contexts with each other compose the context situation in the top level. Despite this point of research is vital, very few studies have insufficiently addressed this issue [32][33][34].To explain this point, let us introduce the following example: in flooding forecasting context-aware system, there are wind speed, temperature, soil saturation, precipitation and rainfall duration which are considered context facts. Rainfall status is an abstracted context, which is concluded using rainfall duration and precipitation context facts. Rainfall status cooperates with the other abstracted contexts such as soil status to compose the final situation of the context, which is the "potential flooding". For this situation, quality of context in the lower level should affect quality of context in the upper level. To the best of our knowledge, this view has not been clearly addressed by the existing solutions of context quality.On the other hand, a context fact could be sensed context such as temperature and user movement, or profiled context such as user calendar, whereas we have derived context in the higher levels. In fact, quality aspects of sensed context should not be the same for profiled context. Thus, these different views for context should be taken into account when thinking about a comprehensivesolution to context quality control. (3) Most existing solutions of context quality parameters did not differentiate between the two basic types of parameters: basic quality parameters, which reflect context validity as a basic quality requirement (e.g. reliability, probability_of_correctness, freshness, and completeness), and the perfection parameters, which reflect other aspects of quality (e.g. privacy, precision, and representation_consistency). We believe that the basic quality parameters should take place, as it could be a basis for many context shortcomings such as context conflicts and context uncertainty.(4) Beside quality parameters that indicate quality level of each piece of context, other general long-term quality indicators are essentiallyneeded to serve different context stakeholders such as context provider, consumer, and CAS developers. To the best of our knowledge, most work that has been conducted by researchers has a lack regarding to this issue. For example, the quality level of a group of context facts in the lower level will introduce an important general indicator to know the quality of the sensor network that produced that context. These indicators could indicate the need to improve the whole network of sensors in a particular area. For example, a network of weather sensors belongs to a particular station that feeds a weather forecast context-aware system.

III. A Frameworkfor Context Quality Assurance in Context Aware Systems

The way to realize context qualityaspects in CASs is to design a broad solution for Quality of Context (QoC), in this paper we introduces a comprehensive framework for QoC. Basically, all quality frameworks should be developed based on the concept of quality management systems (QMSs). QMSs are quality systems developed in business field to ensure the quality level of products or services.A QMS is a complete system that includesa collection of business processes that realize the admitted qualitypolicy and quality objectives [40][41]. In the same way, the proposed framework should encompass the following component: the quality objectives or parameters, a quality policy based on the context nature, and control processes that will enforce qualitypolicy.

This section startsby introducing the idea of context layers that was the basis of our framework where the proposed quality parameters and quantification methods are based on this idea. Then, our new proposed definition of QoC is introduced. This definition also affects the development of framework elements. Finally, the different constructs of our framework are described.

A. Multilayer Context Model

The basic idea behind building the proposed framework is the *contextmultilayers*. This idea affect the designing of different framework construct. Actually, using contextmultilayers to conclude the quality level of context is not a new idea. In [32][34], authors introduced a model for contextlayers to compute the context confidence. The idea was to introduce three layers of context, start in the bottom with context facts thatareproduced by sensors and then the intermediate layer that contains the abstractedcontext, which is aggregated from many contextfacts or abstracted from a context facts.The top layer contains the contextsituation, which is the final derivedcontext.

In our proposed framework, we found the idea of contextmultilayers is necessary for the following reasons:

(1) Context as a nature is multilayered; there is a sensed context, profiledcontext, and derivedcontext. Sensed context is a simple fact, which is collected/acquired using sensors such as temperature degree, hurt pulses or wind speed. Profiled context is a context, which is assigned by the user such as meeting dates or user calendar. Derivedcontext means there are underlying context that the situation is derived upon it [13][17]. Unfortunately, most work achieved so far for quality parameters focusedonly on sensedcontext whereas CASs make their decisions and services usually based on derived context.

(2) Some parameters are not applicable for some layers. For example, probability_of_correctness is a vital parameter that reflects the extent that a piece of contextelement is valid according to the occurrences of other pieces of context based on the previous history. This parameter is applicable for the context in the lower level where the pieceofcontext is not derived and there are real occurrences and a previous history for this pieceofcontext and for the other pieces of context. The complete view for quality parameters that is suitable for each layer is introduced in our parameters model in Section.

So, distinguishing between context layers and consequently context parameters for each layer is necessary.

The proposed multilayered context model includes three basic layers: contextelements layer, contextobjects layer, and contextsituation layer. The quality of context for each layer depends on the quality of underlying layers. The definition of the three concepts from inner to outer layer is below:

***Contextelements layer*:** this layer contains the contextelementswhich are basically captured from sensors or profiled from a user such as temperature, movement, location, humidity, and a meeting date in a user calendar. This layer is the main difference between our work and the layers proposed in [13][17] as the authors there consider only the sensed context in this layer.

***Contextobjects layer*:** this layer contains the context entities that are derived using underlying context-elements. Also, it can be derived using other underlying context-objects. Examples of derivedcontext are weather status, flooding status, and user status. This intermediate layer could contain inner layers but the context in this layer does not represent the final target context situation. The contextobject could be e.g. a person, a place, or an event.

***Contextsituation layer*:** this layer is the same as contextobjects layer, as it is a derivedcontext, however it represents an independent layer to distinguish it as a final target context where the services of context-aware systems depend on directly.

Figure 1 illustrates the nested layers of context. Our parameters model and also the evaluation methods are designed based on these nested layers as clarified in Sections and.

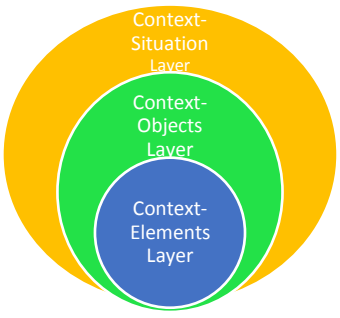


Figure 1 Nested Layers of Context

B. A Generalized Definition for QoC

Referencing to the quality issue in general, specialist can use some verified heuristics to ensure conformity of required quality level when dealing with computing systems. These heuristics are always represented as data to facilitate computing quality realization and management in order to support the performance automatically. As a result, quality of context is considered as information about the context information that enables us to judge the quality level of context. In light of that, the first definition of QoC was introduced by [1] as:

"Quality of Context (QoC) is any information that describes the quality of information that is used as context information".

Later, the definition of QoC is modified by [2] to involve the subjective nature to the concept with engaging user satisfaction to the definition:

"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".

In our view, the definition which is proposed by [1] was simple as it cannot capture all and critical quality aspects that affect and ensure quality of context. This definition focuses on the representation of quality more than the key quality aspects for context within the context-aware ubiquitous environment. This definition of QoC guides researchers to elaborate the context quality as general and with concentrating on inherent objective information about context information apart of consumer view and the real context. Thus, this definition is modified by [2] to involve the subjective nature to the concept with engaging user satisfaction to the definition. This view is more close to the quality references models and its nature. In addition, both earlier definitions concentrate on sensed context despite the context can be sensed, profiled, or derived.

Our definition of QoCis as follows:

"The degree of conformity of the context to represent the situation in the environment and to fulfill the requirements of the context consumer".

In this definition, quality of context is not limited to the sensed data compared to the definition that has been proposed by [2]. The contextsituation that is mentioned in the definition is the context in higher level. It is derived using sensed and profiled. So we can that all layers of context are included in the definition. In addition, credibility of context situation measures the extent that the context represents the real word. This definition indicates that the*Credibility*parameter should be considered as an important quality parameter for context. In addition, it indicates importance of realizing the consumer requirements.

These implications, which are inspired by the new definition, form the basis of other quality framework constructs**.**

C. Architecture of QoC Framework

This section introduces the architecture of the proposed framework. In the heart of the framework, there are two important constructs: the quality policy and the control processes that enforcesthe quality policy.To definequalitypolicy, quality parameters should be determined. Therefore, the proposed frameworkbasically encompasses quality parameters, qualitypolicy, and quality control processes as shown byFigure 2.

The main parts of the MCFQoC framework are:

- The conceptual model, which represents the theoretical basis for other components; the conceptual contains the following components: (1) Multilayered context model, (2)QoC parameters, and (3) QoC parameters evaluation methods.
- Quality assurance system, which contains three main components: (1) Quality policy, (2) The run-time quality control processes which include two sub processes: IPQP algorithm for handling context shortcomings and CAS provision with QoC parameters for supporting context derivation, and (3) QoC long-term general indicators.

The next sections describe the these parts in details.

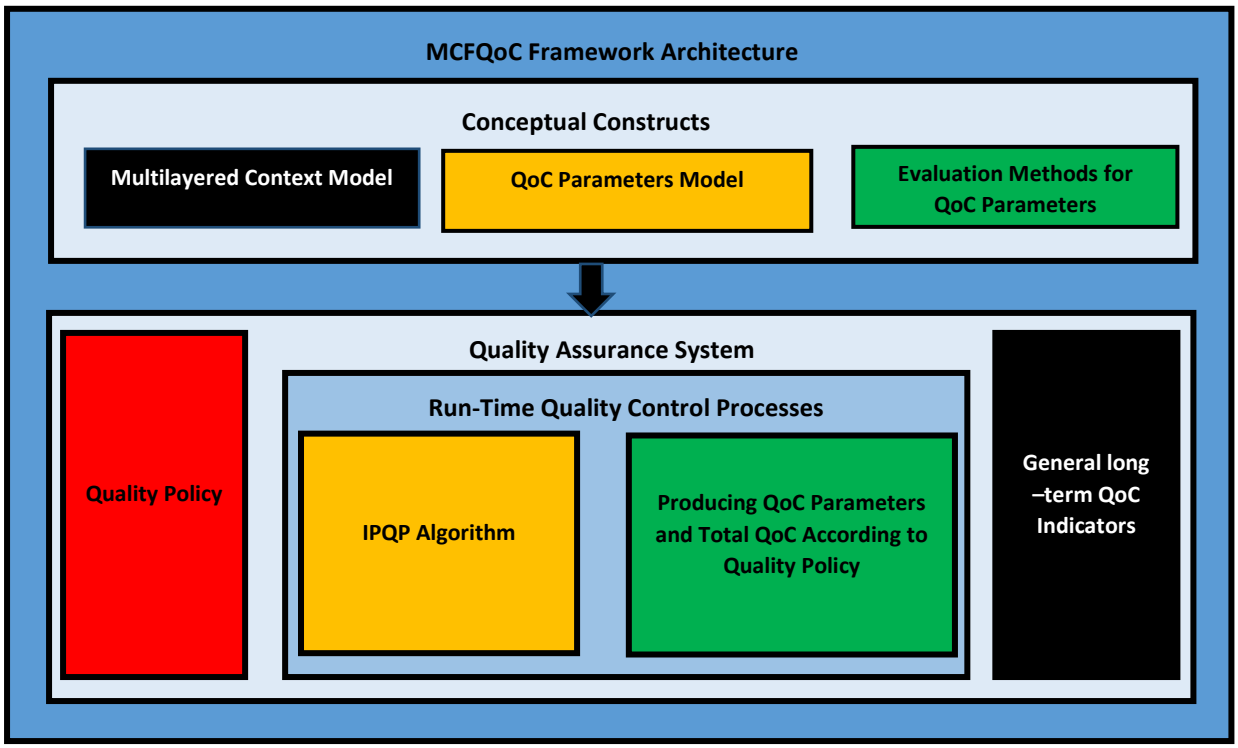


Figure 2MCFQoC Framework Architecture

D. QoC Parameters Model

Designing the model of QoC parameters was depending on the idea of context multilayers and context taxonomy. Quality parameters model includes a package of parameters. Many of them have been already addressed by researchers; however, in this research we have proposed a new parameter and also a new distribution for the parameters according to the context level. Figure 3 introduces a summary of quality parameters for each context layer. In general, there are nine quality parameters, which are reliability, timeliness, completeness, credibility, accuracy, usability, representation_consistency, security_level, and probability_of_correctness [1][2][3][7][8][11][30][31][47]. These nine parameters combine most quality aspects mentioned in the literatures without repetitions.*Credibility* parameter is a new parameter that we proposein light of the QoC definition to reflect the reality of a context. It is a deep parameter, as it combines reliability, timeliness, completeness, and probability_of_correctness all together to represent the context *reality*. We think that we need one parameter to reflect the reality of context. We could not trust the reliability parameter to do that because reliability depends on sensorsthat are prone to malfunctions. So, probability_of_correctness parameter can supports the reality of context beside the reliability parameter. Especially for contextsituation, the recommended quality parameter is Credibility. That is because the main objective of all quality parameters in the lower layers is the credibility of the final context situation as clarified by our QoC definition.

We define *Context Credibility* as follows: "*To what extent a piece of context reflects the reality*".

As clarified by Figure 3, the parameter probability_of_correctness is applicable only for context in contextelement layer. That is because it reflects the conformity of all contextelements for the contextelement under investigation based on the real occurrences in the previous history. Therefore, we cannot apply this parameter to the derivedcontext, as we need the occurrences in the previous history, which are not available for the derivedcontext.

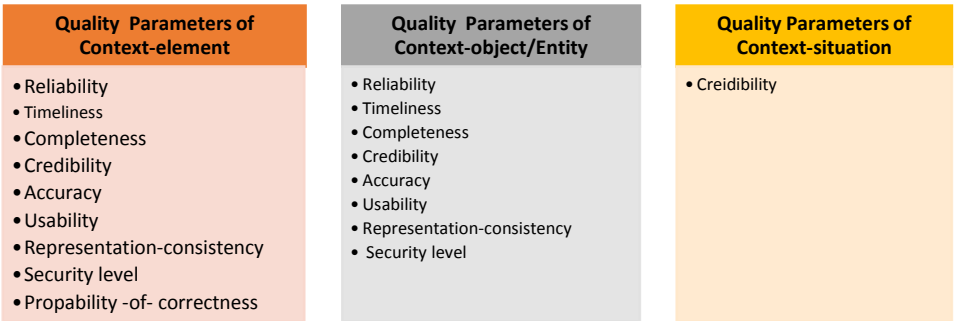


Figure 3 The Proposed Model for QoC Parameters

E. Evaluation of Quality Parameters

This section describes the new formulas, which are used for calculating some quality parameters that are included in our model. Many parameters in our proposed model are calculated according to the formulas mentioned in literatures as shown in Table 1.

Modifications that are carried out upon the calculation methods encompass main three classes: (1) the modifications that have beencarried outfor the formulas which are used to calculate some quality parameters to make them more accurate for different contextlayers and types. These parameters are reliability, timeliness, completeness, and evaluation of profiledcontext. (2) The modificationsthat are related to usability parameter by introducing a new formula to calculate it. (3) For *Credibility* parameter, new formulas were introduced to calculate credibility in different contextlayers.

Reliability and Timeliness Evaluation for DerivedContext:

The formulas that were introduced in literatures for computing reliability and timeliness were applicable only to sensedcontextelements in the lower layer of context as a metadata used to calculate the whole value belongs to a single contextelement. For example, the metadata values, which are used to calculate reliability, include sensor accuracy, distance between sensor and the target, and maximum valid distance. All these previous metadata values belong to a single sensedcontextelement. Obviously, these metadata values are not applicable to the derivedcontext as the derivedcontext is composed from many single pieces of context. In this case, reliability and timeliness will be calculated as a total reliability and timeliness of contextelements that compose the contextobject taking into account the relative weight of contextelements for composing the derivedcontext as shown byformulas 1 and 2 and .

$$Reliability(derived_context) = \sum_{i=1}^{n} reliability(piece_of_context_i) * weight(piece_of_context_{i_i}) \quad (1)$$

$$Timeliness(derived_context) = \sum_{i=1}^{n} timeliness(piece_of_context_i) * weight(piece_of_context_i) \quad (2)$$

Where *n* is number of piecesofcontext that compose the derivedcontext.

Completeness Evaluation:

Our formulathat introduced to include derivedcontext that is composed fromsimplecontext or even of other derived piecesofcontext as shown by Formula 3.

$$Completeness(derived_context) = \sum_{i=1}^{m} weight\ (piece_of_context_i) * completeness(piece_of_context_i) \quad (3)$$

Where*m* is the number of available pieces of context that compose the derivedcontext.

For simplecontext in the lower level, the completeness of that context would be always **1** if it is available or **0** if not available. Formula 4 shows the completeness. This formula is applicable forsensed or profiled context.

$$Completeness(simple - context) = \begin{cases} 1 & : if \quad simple_context\ is\ available \\ 0 & : if\ simple_context\ is\ not\ available \end{cases} \quad (4)$$

Usability Evaluation:

To judge if a piece of context is usable or not, it should satisfy two requirements: (1) accuracy requirement where it should be at the level of required accuracy for use; (2) format/representation requirement as it should be in the required format. Formula 5 has been introduced to involve the two aspects to calculate the usability:

$$Usability\ (piece_of_context) = wA\ .\ (Accuracy\ (piece_of_context)) + wR.\ (\ Representaion_consistency\ (piece_of_context) \quad (5)$$

Where,wA and wR are the relative importance weights for the accuracy parameter and the representation_consistency parameter respectively. These weights are set to 0 if the application does not need the corresponding parameter. Accuracy is calculated as in Formula 7 (Table 1)[7]; based on that, the representation_consistency is calculated using Formula 6 :

$$Representaion_Consistency(derived_contexxt) = 1 - \frac{current_transformation\ _cost}{maximum_transformation\ _cost} \quad (6)$$

Formula 6 is adapted from [2], where the parameters current_transformation_cost and maximum_transformation_cost are determined by user to reflect the effort of transforming the context from one format to another.

Credibility Evaluation:

Context *Credibility* is aproposed parameter to reflect the extent to which a pieceofcontext is close to reality. The greater the credibility value, the moreconfidence ofthecontext to be closer to reality.

Calculatingcredibility in different contextlayers ould be different. We will start with credibility for contextelements. To calculate credibility, three aspects can be used: reliability, timeliness, completeness (for derivedcontext), and probability_of_correctness as shown by Figure 4.Probability_of_correctness is included in the Formula8 with relativeweight of 50%, which is equivalent tothe total weightoftheother parameters. This large weight is assigned to overcome the fact that the parameters timeliness and completeness are subjective where the user determines some of their metadata values. In addition, reliability parameter depends on sensors, which are prone to faults. Probability_of_correctness depends on the affirmation of other piecesofcontext based on the previous history of a context.

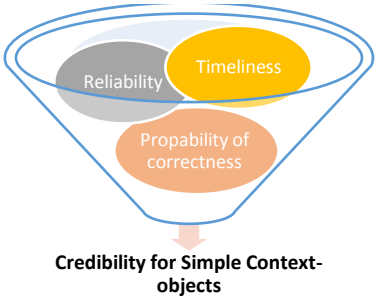


Figure 4 How Credibility is AComplex Parameter

Hence, probability_of_correctness parameter is included to make balance and to make credibility parameter more objective (Figure 5). This mixture of parameters should give a good indication of context credibility. Based on what mentioned above, Formula 7is proposed for piece_of_context credibility evaluation.

$$Credibility(\text{simple_context}) = Avg[Avg[\text{Reliability(simple_context)}, \text{Timeliness(simple_context)}],\ \text{Probability_of_correctness(simple_context)}] \quad (7)$$

For a derivedcontext, credibility parameter is calculated as a total credibility of all piecesofcontext that compose the derivedcontext. Formula 8 has been proposed for the derivedcontext:

$$Credibility(derived_context) = \sum_{i=1}^{n} credibility(piece_of_context_i) * completeness(piece_of_context_i) * weight(piece_of_context_i) \quad (8)$$

Where *n* is the number of piecesofcontext, which compose the derived context.

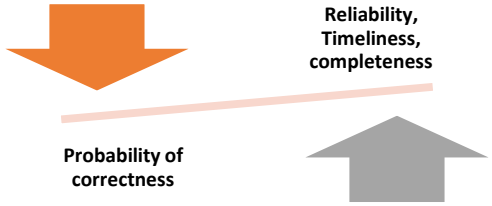


Figure 5 Howredibility Parameter Makes a Balance between Objectve and Subjective Quality Aspects

Quality Parameters Evaluation for ProfiledContext:
For a simple profiledcontext, which is usually assigned by user, few researches investigated methods to evaluate its quality parameters. For example, profiledcontext should be updated periodically by user to insure freshness of the context and the timeliness parameter should reflect that. In addition, similar to the sensedcontext, the accuracy level of a profiledcontext should be in the requiredlevel according to CAS's needs. Referring to all quality parameters in our model, the profiledcontext is subjected to most calculating methods of parameters with one exception. The reliability of a sensedcontext depends on the sensor reliability which is determined by the manufacturer, and the distance between sensor and target. Whereas the reliability of aprofiledcontext is always 1 as this value is recorded by the user [4][35].

Table 1 summarizes the formulas used in our model to calculate the different quality parameters.

Table 1 A Summary of Formulas Used to Evaluate the Quality Parameters in the ProposedModel

Formulas	Reference
Reliability	
$\boldsymbol{Reliability\ (sensed-simple-context)} = \begin{cases} \mathbf{1} - \frac{\boldsymbol{d(s-\varepsilon)}}{\boldsymbol{d_{max}}} * \boldsymbol{\delta} & : \ \boldsymbol{if} \quad \boldsymbol{d(s-\varepsilon) < d_{max}} \\ \mathbf{0} & : \ \boldsymbol{Otherwise} \end{cases}$ $\boldsymbol{where\ d(s-\varepsilon)}$ is the distance between the contextobject and the sensor, $\boldsymbol{d_{max}}$ is the maximum distance and $\boldsymbol{\delta}$ is the accuracy of the sensor. $\boldsymbol{Reliability(derived_context)} = \sum_{i=1}^{n} \boldsymbol{reliability(piece_of_context_i) * weight(piece_of_context_i)}$ Where *n* is the number of piecesofcontext that compose the context. $\boldsymbol{Reliability(profiled_simple_context) = 1}$	[2] New New
Timeliness	
$\boldsymbol{Timeliness\ (simple_context)} = \begin{cases} \mathbf{1} - \frac{\boldsymbol{Age(simple_context)}}{\boldsymbol{Valdityatime(simple_context)}} & : \ \boldsymbol{if} \quad \boldsymbol{Age(simple_context) <} \ ValidityTime(simple_c \\ \mathbf{0} & : \ \boldsymbol{Otherwise} \end{cases}$ $\boldsymbol{Age\ (simple_context) = T_{curr} - T_{meas}(simple_context)}$ Where$\boldsymbol{T_{curr}}$ is the current_time, and $\boldsymbol{T_{meas}(simple_context)}$ is the *measured_time* for $\boldsymbol{simple_context}$ and *validity_time* is determined by the context consumer. $\boldsymbol{Timeliness(derived_context)} = \sum_{i=1}^{n} \boldsymbol{timeliness(piece_of_context_i) * weight(piece_of_context_i)}$ Where *n* is the number of pieces_of_context that compose the context.	[2] New
Completeness	
$\boldsymbol{Completeness(derived-context)} = \sum_{j=1}^{m} \boldsymbol{weight\ (piece_of_context_i) * completeness(piece_of_context_i)}$ Where *m* is the number of available pieces of context. $\boldsymbol{Completeness(simple_context)} = \begin{cases} \mathbf{1} & : \ \boldsymbol{if} \quad \boldsymbol{simple_context\ is\ available} \\ \mathbf{0} & : \ \boldsymbol{if\ simple_context\ is\ not\ available} \end{cases}$	New New
Representation_Consistency	

$\boldsymbol{Representaion_consistency}\ (\boldsymbol{piece_of_context}) = \mathbf{1} - \frac{\boldsymbol{current_transformation_cost}}{\boldsymbol{maximum_transformation_cost}}$	New
Accuracy	
$\boldsymbol{Accuracy}\ (\boldsymbol{piece_of_context}) = \frac{\boldsymbol{current_granularity_level}}{\boldsymbol{total_number_of_granularity_level}}$ ***(7)***	[7]
Usability	
$\boldsymbol{Usability}\ (\boldsymbol{piece_of_context}) = \boldsymbol{wA}.(\boldsymbol{Accuracy}\ (\boldsymbol{piece_of_context})) + \boldsymbol{wR}.(\boldsymbol{Representaion} - \boldsymbol{consistency}\ (\boldsymbol{Opiece_of_context}))$	New
Security_Level	
$\boldsymbol{Security_level}(\boldsymbol{piece_of_context}) = \frac{\boldsymbol{current_security_level}}{\boldsymbol{total_number_of_security_level}}$ Where $\boldsymbol{current_security_level}$ and $\boldsymbol{total_number_of_security_level}$ are determined by user.	[7]
Probability_of_Correctness	
$\boldsymbol{Probability_of_correctness}(\boldsymbol{simple_context}) = \sum_{i=1}^{m} \boldsymbol{confidence}(\boldsymbol{simple_contexti} \Rightarrow \boldsymbol{simple_context})$ $\boldsymbol{Confidence}(\boldsymbol{simple_contexti} \Rightarrow \boldsymbol{simple_context}) = \frac{\boldsymbol{sigma}(\boldsymbol{simple_contexti} \cap \boldsymbol{simple_context})}{\boldsymbol{sigma}(\boldsymbol{simple_contexti})}$ Where $\boldsymbol{simple_context}i$represents other pieces of context which are captured with the $\boldsymbol{simple_context}$.	New
Credibility	
$\boldsymbol{Credibility}(\boldsymbol{simple_context}) = \boldsymbol{Avg}\begin{bmatrix}\boldsymbol{Avg}[\mathbf{Reliability}(\boldsymbol{simple_context}), \mathbf{Timeliness}(\boldsymbol{simple_context})], \\ \mathbf{Probability_of_correctness}(\boldsymbol{simple_context})\end{bmatrix}$	New
$\boldsymbol{Credibility}(\boldsymbol{derived_context}) = \sum_{i=1}^{n} \boldsymbol{credibility}(\boldsymbol{piece_of_context}_i) * \boldsymbol{completeness}(\boldsymbol{piece_of_context}_i) * \boldsymbol{weight}(\boldsymbol{piec_of_context}_i)$ Where *n* is the number of piecesofcontext, which compose the derivedcontext.	New

F. Context QualityPolicy

In practice,the weight of a quality parameter varies depending on the nature of the CAS application [42][43][44][45].In our framework, the quality policy will containthe quality parameters that are required for the CAS and the weight assigned to eachparameter. The idea of determining different quality parameters with different weights for different CASs is not new (e.g., authors in [7] and [14] have addressed such idea). However, the contribution of this research is the way that determines the quality criteria in addition to the way of assigning the relative weights. Our approach is based on AHP method to produce the parameters weights. An overview of AHP is introduced below.

The total quality of context has been addressed in literatures and it is evaluated using the admitted quality policy for each piece of context. The most common way is that proposed by authors in [14] which introduced the idea of assigning weights for each QoC parameter to use them for calculating the total quality of context. This way uses a predefined 7-level scale to assess the parameters weights.

In our framework, we use AHP to assign the parameters weights, the total quality of context will be calculated using Formula 9.

$$QoC(piece_of_context) = \sum_{i=1}^{n} wi * qi \quad (9)$$

Where n is number of quality parameters, q is quality parameter, and w is parameter weight.

An Overview of AHP:

The Analytic Hierarchy Process (AHP) [23] is a decision-making method developed in the late 1960's by Thomas Saaty, one of the pioneers of the operational research field. This method is characterized by

its ability to incorporate qualitative factors using quantitative variables within the evaluation process of the different decisions alternatives [23][24][25][18]. AHP is used to assign quality parameters as a hierarchy for each contextlayer similarly to decisions analysis according to AHP. AHP uses a hierarchy of criteria that affect the final decision. In addition, based on AHP, the parameters weights are determined based on accurate pairwisecomparison for each criteria and then the final weight for each parameter is calculated taking into account the results of the pairwisecomparison. The pairwisecomparison is useful in the case that the number of criteria is large. Assigning the weights requires comparing each criterion to all other criteria to assign an accurate weight.
In our proposed quality parameters model, the number of quality parameters is nine. If all these parameters should be realized for a CAS, determining the weights will be slightly hard and could be inaccurate. AHP as a method is reliable. It achieved remarkable success for a wide range of decisions [23][24][25][18].

G. Context QualityControl Processes

According to ISO 9000 definitions [22], quality control is "a part of quality management focused on fulfilling quality requirements" [22]. Accordingly, context quality control is a process that works to fulfill the context quality requirements. The context quality requirements are not only the validity of context values, they also contain many other quality attributes such as security level, timeliness, etc. Context quality requirements differ from a CAS to another based on the problem nature and context type. Each CAS has a different "quality policy" that determines quality attributes, which should be realized in the context along with the important weights for each attribute.This quality policy is enforced by the control processes. In this research, we propose two control processes as follows:

The First Control Process: Providing CASs with Quality Information for Resolving Context Uncertainty

This process produces the different quality parameters and total quality, and feeds a CAS by these values to improve the context provesion. The CAS then compares these values to predefined thresholds before using the associated context in decision making. This control process is not completely new however, the way of calculating the total quality of context and assuming this process as a construct within the framework is the contribution of this research.

The Second Control Process:IPQP Approach for HandlingSensed Context Shortcomings

As a part of our framework, we introduce a quality control process to improve context acquizition automatically. This control process has been designed for handling context shortcomings in run time. In this process, the proposedframework apply our approach *IPQP(Integrating Prediction with context Quality Parameters*). IPQP is extension for our previous work RCCAR algorithm[52]for resolving sensed context conflicts. IPQP handles some basic shortcomings of sensed context according to quality aspects. Three types of context shortcomings are addressed: context missing values, context erroneous values and context conflicts.
IPQP include two basic steps for handling context conflicts. The first step is applying RCCAR to recognize the valid values among many conflicted context values, and then apply a new step that uses context quality aspects to select the best value among the valid values. Invalid values can be defined here as the values that are far from the correct value. TOPSIS method (The Technique for Order of Preference by Similarity to Ideal Solution) [41] is used to select the context value that has the greatest quality level after excluding the invalid values. For handling missing and erroneous values, we use the only first step by using RCCAR for predicting the required value.

Our approach RCCAR (Resolving Context Conflicts Using Association Rules) is introduced and published in [52]. RCCARexploits the pervious history of a context to predictwhat among the context conflicted values are valid and what is not. The prediction uses Association Rules (AR) technique [17][21] to get all associations that combine context elements together based on the occurrences in the previous history. AR definitely discovers what goes together in data based on data occurrences in database. More explanations and details could be found in [52].

The Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) [26] is a multi-criteria decision analysis method. A decision problem is described by a set of alternatives; each one is evaluated against a set of weighted criteria. TOPSIS method is based on the view that the best alternative is that has the shortest geometric distance from the positive ideal solution and the longest geometric distance from the negative ideal solution [26][27][28][29]. In our case, alternatives are the

context conflicted values, and criteria are simply QoC parameters which represent quality policy; the goal is to determine the value that has the best quality for a context.

IPQP Algorithm

This section describes how IPQP handles the different shortcomings of a context in order to control the quality level of context. For missed and erroenous context values, IPQP predicts an alternative *valid* value for the context. However, for context conflicts, there is a chance to introduce better solution by involving different quality aspects beside validity to select the best valid value among many conflicted values. IPQP algorithm receives context value(s) and recognizes the type of shortcomings (if any) before starting processing.

- **Handling Context Missing Value**

In case that the context value is missed, the solution introduced by IPQP is predicting the missed value using RCCAR approach. Actually, RCCAR was proposed to resolve context conflicts by computing the affirmation of context elements (probability_of_correctness) for each conflicted value of an investigated context element, and then choosing the value that has the greatest affirmation. However, the purpose of the case in hand is slightly different.In this case, RCCAR is used to compute the affirmation for all values that the context history includes , then the value that has the greatest affirmation will be selected.

- **Handling Context Erroneous Value**

The erroneous value is defined here as the value of context which is very far to the valid value. IPQP decides that the context value is erroneous if its Probability_of_Correctness (PoC) using RCCAR is less than the predefined threshold. This case is rare as the context value is usually sensed using many sensors. However, if this case happens, the solution that IPQP introduces is the same for the missing value case. IPQP uses prediction values for the context value under estimation based on the previous history of context. RCCAR has been employed to compute the affirmation for all expected values according to the previous history and then, to select the value which has the greatest value.

- **Resolving Context Conflicts**

In this case, context includes many values for one context element, some of them could be erroneous and the others are valid values with different quality levels based on the admitted quality aspects (parameters) for a CAS. IPQP should select the valid value, which has the greatest quality level among many conflicted values. Figure 6 illustrates the different stages for resolving context conflicts using IPQP.

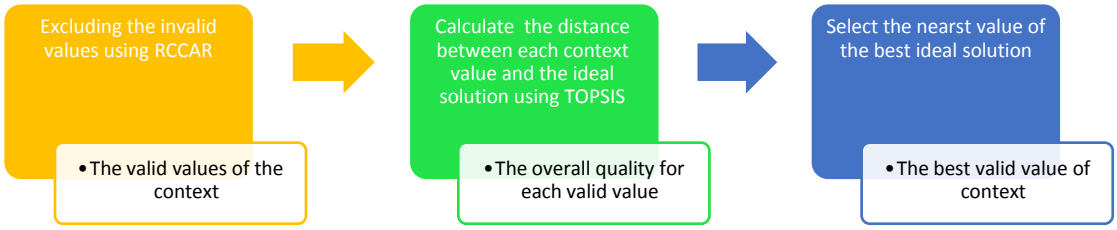


Figure6 Main Stages of IPQP Algorithm for Resolving Context Conflicts

Stage One: **Excluding the invalid values from conflicted values using RCCAR**

In this stage, IPQP excludes the erroneous values from conflicted values. These values are invalid because they have a large distance between real and predicted values. The method that is used to predict the valid values is RCCAR as mentioned above. In IPQP, RCCAR is used only for excluding invalid values from different conflicted values; and then choose the best value according to quality aspects using TOPSIS. This is because that we should take into account the CAS's context nature and admitted quality policy when resolving conflicts. RCCAR calculates the total affirmation (probability_of_correctness) of each conflicted value. The value with the highest total affirmation is selected, as well as all values with the permitted distance from the best value; for example, the difference between 10^0C and 11^0C for temperature is fine. Context user determines the value of permitted distance for each context element according to the context problem. The selected values will be processed further in stage two.

Stage Two: **Calculate the distance between each context element value and the best ideal solution using TOPSIS and the quality policy**

In this stage, we have used the well known method TOPSIS in our approach as multi-criteria decision analysis method as described in Section. TOPSIS algorithm produces the distance between each valid context value and the best ideal solution according to the quality policy.

Stage Three: **Select the context with best value**

In this stage, the context value that has the shortest distance to the ideal-positive-solution and the longest distance to the ideal-negative-solution will be selected as the best value for the corresponding context values.

H. General Quality Indicators

Quality parameters presented in Section represent quality aspects for each piece of context at different levels of context. Besides that, a general quality indicators will be important for the context levels to find out the total quality of acontext in different layares and to reflect some general quality aspects. There are no new parameters introduced here; only general quality indicators for the different layers derived from the parameters values of different pieces of context in each layer. For example, the general quality level of a sensedcontext which is produced by a sensor network belongs to a certain party can be considered an important indication to assess the performance of a sensor network as a whole and can help planning for future improvements.

Actually, these indicators are useful for future and do not serve the daily work of CASs as the quality parameters do. If these general indicators are monitered over the time, they will contribute in improving a context design and resource planning. In general, data mining techniques can be used to detect the abnormal patterns in context quality indicators over the time and provide the decision makers by valuable information to think and improve the whole system and networks that used for capturing the context in CASs.

MCFQoCframework adopts different types of quality indicators to expand the utility of quality results in order to cover wide range of usage. These indicators can be summarized as follows:

1. **QualityDegree of each Contextlayer**: This general indicator reflects the general qualitylevel of all pieces of context in each contextlayer based on the quality policy. This indicator is important for the context producers and CAS's developer to know the general level of quality for each layer. This can help to focus on the waek layer and investigate the reasons behind that.
2. **Average of the Credibility for each Layer**: this indicator reflects the context reality in each contextlayer. It could especially lead the context producer to improve the context derivation in future. For example, if the first layer in the bottom has low quality level, this could lead the context provider to think how to improve the sensor network. If the context level in the lower level is fine but the quality goes down when we get up through contextlayers, this could lead to think about the pieces of context that have the most relative weight and how to improve the root context information for them.
3. **Average of the Non-Validity Parameters for each level**: this indicator reflects the perfection level based on non-validity parameters such as security and representation_consistency according to the qualitypolicy. We prefer to separate these parameters of the credibility parameter to monitor them apart of the credibiliy as the purpose of them is different. The purpose of the credibility is to measure context validity and support derivation for the context with managing context uncertainity, wheras the purpose of non-validity parameters is to measure the system perfection. It is important to be aware about the changes of different aspects of quality and interpret the reasons behind that to keep quality perfection of the context in a fine level.
4. **Averages of all Quality Parameters in each Context Level**: these indicators are very important to let the context producer aware about each quality parameter in different layers ecpeically if there is a need for that. For example, if a big defect is remarked for non-validity parameters in such contextlayer and then the analyzers found that the problem is definitly due to the securitylevel, they could review the autherities that are assigned for different context consumeres, as they could give many access rights to the context during sytem operation without studying that carefuly.

I. How to Use MCFQoCFramework

To use MCFQoCframework, we propose the following steps:

1. **Context Analysis**: this step concernswith determining the context levels according to the context structure in terms of context element, context objects, and context situation.
2. **Determining the QualityPolicy:** the user, context consumer, application developer and all other stakeholder should conduct this step. This step contains two sub-steps:

a. **Determining the QualityParameters:**according to the nature of theapplication, stakeholders should carefully determine the quality parameters, which are appropriate to the context based on the parameter model that is introduced in the framework.
b. **Determining the Weights for the Quality Parameters:** in this step, stakeholders determine the important relative weights for each quality parameter according to the nature of the application and their expertise in the field. To be more scientific in this step, we employ AHP as described in Section. To accomplish this step automatically, there are many tools available for AHP. However, this part forms a basic component of our prototype that is developed to test the whole framework.

3. **Applying the QualityControl Processes:** this step enables a CAS to realize the quality policy.Actually, this part should be achieved automatically in run time.
4. **Monitoring and Studying the General Indicators**: this step is achieved by developing CASwith some services and a simple dashboard containingthe general quality indicators.

IV. IMPLEMENTATION

When examining ubiquitous computing applications, realistic testing environments will be essential. However, creating a realistic testing environment is highly costly. Rather, prototyping techniques are used to test CAS applications [46].This section presents the implementation of a prototype of the proposed framework. It depictsthe architecture of the developed prototype, and the data that is used to test the framework inareal-worldcase study of flood forecast system. Basically, MATLAB has been used to develop the prototype. This is because most framework computations are based on arrays, which are easy to implement using MATLAB.IPQP as a vital computing component is implemented using the WEKA tool; as it provide most known algorithms of association rules that are used in IPQP and RCCAR.

A. The Prototype Functionalities

The developed prototype aims at implementing and evaluatingthe proposed framework. A prototype has been implemented for the main constructs that need computing; definitely, assigning the quality policy using AHP, evaluating QoC parameters, calculating the overall quality using TOPSIS, and applying the qualitycontrol processes. The only construct that is not implemented in the prototype is the general quality indicators as they are not required for daily CAS's work as they are a long term indicators monitored over the time.To enable the prototype doing its work, it should be fed with the context data and metadata. In addition, the prototype requires determining quality parameters by the user and doing a pairwise comparison between each pair of parameters to compute the relative weights for each parameter used in the prototype.

Figure 7 shows the prototype architecture. The prototype contains four main modules with data repository as most modules get/put their data from/to this data repository. Data repository is represented in the prototype as simple Excel files where the prototype reads/writes from/to them. The description for each module's functionality can be presented as follows:

(1) Context Data and Metadata Configuration Module

This module is responsible for receiving the context data and metadata and keeping them available for other modules. The context data and metadata are context hierarchy and context values with the associated metadata valuesfor each quality parameter.

(2) QualityPolicy Configuration Module

The main functionality of this module is enabling the user to assign the required quality parameters and let user do a comparison for each pair of parameters; then this module will produce the relative weights for each parameter accordingly. If the pairwisecomparison results is not consistent, the prototype informs user that he/she should re-doing the pairwisecomparison.

(3) IPQP Module

This module is responsible for applying IPQP algorithm for context quality control. Some steps of IPQP has been achieved with aid of Weka tool and the results are loaded to the prototype as input files. The part that achieved using WEKA is the association rules algorithm for prediction.The version used in this research is WEKA 3.**7.7**.WEKAsupports different platforms, includes various algorithms, and accepts a various types of data input[20].

Two real datasets were used for IPQP experiments. These two datasets are available for researchers to examine their solutions:Southampton monthly weather historical data from the year 1855 to 2000 [19], and Cardiff climate data for five days [21].Southampton dataset is a historical data for some

weather variables. This data is officially collected and recorded by Southampton Weather Station and published by its Website [19]. This data set contains 1744 instances. The collected weather data is recorded as instances, each instance contains values of some weather variables, which are shown in Table 2.

Table 2 Variables of Southampton Weather Historical Dataset

No.	Variable	Description
1	YYYY	The year
2	MM	Month
3	Tmax	Temperature Max (Avg)
4	Tmin	Temperature Min (Avg)
5	AF	Air Frost
6	Rain	Rainfall
7	Sun	Sunshine Hours.

Cardiff Climate Datasetis provided on the Internetby The Met Office which is the UK's National Weather Service organization. It has a long history of weather forecasting as it has been working in the area of climate change for more than two decades [21]. This dataset is used to examine how the prediction is accurate according to five days (short history) as climate does not change accidently, and within these five days, several readings were performed using different stations and sensors. According to this close history, IPQP has to predict accurately the valid values for different variables for the next new reading. The variables that are available in this dataset are shown in Table 3[19].

Table 3 Variables of Cardiff Climate Dataset

No.	Variable
1	Date
2	Time
3	Precipitation Probability (%)
4	Humidity (%)
5	Visibility
6	Temperature (C)
7	Wind speed (mph)

(4) Quality of Context Parameters and Indicators Producing Module

The function of this module is to compute the total quality for each piece of context starting with the contextelements level and ending by the contextsituation level. This module represents the second control process in the proposed framework beside IPQP module.

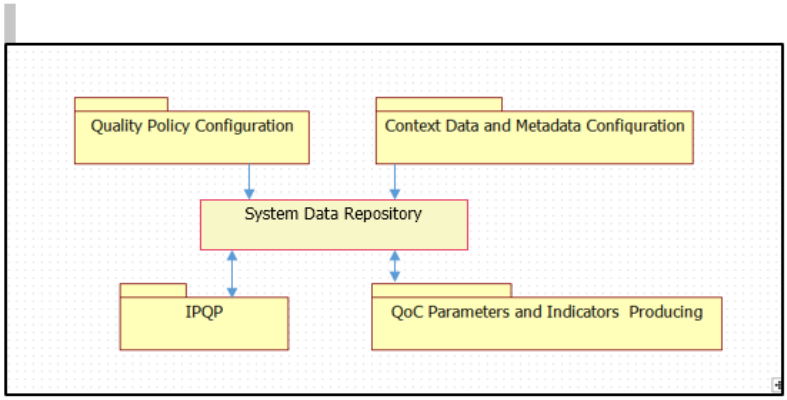


Figure 7 The Prototype Architecture

B. The Description of the Case Study

The case study that is used for framework evaluation is a true flooding case within a CAS for flooding forecasting. There are many causes for flooding, some of them are natural based and the others are

human based. Our case study addresses the natural based causes and changes for flooding. The context contains some variables about weather and soil. These variables together monitor possible situations of flooding. Flood forecasting system usually uses real-time precipitation and stream flow data to forecast flow rates and water levels for periods ranging from few hours to days ahead depending on the size of the watershed or river basin [38][39]. Our case study is true potential flooding situation, however the context has some shortcomings that affect the whole quality of context and then could lead to wrong context derivation.
The framework will introduce different quality indicators about the context and let the system makes decisions accordingly.
The context items include three levels of context as it is in real word: context elements which are raw/sensed context, context objects which can be sensed or derived, and finally the context situation which is the final level of derived context. Figure 8 clarifies the hierarchy of the context. It shows the selected context variables in each context level and the synthesis relations between these variables through the different levels.

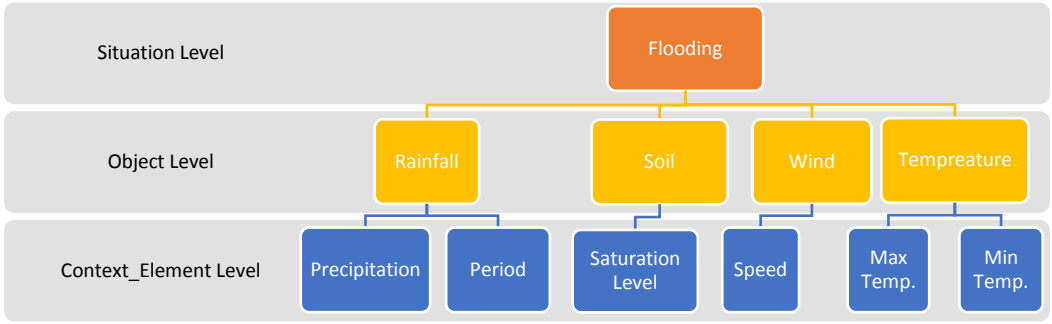


Figure 8 The Hierarchy of Context

The values of context variables and the metadata for all variables have been introduced in this section. Metadata are used for calculating the different quality parameters. The context data for each level starts with lower context levels and goes to higher levels. Table 4 introduces the context data for context element level.

Table 4 The Context Data for Context Elements

Precipitation *mm*			**Period** *hrs*	**Wind Speed** *km/hrs*	**Saturation Level** *VSW%*	**Max Temp.** *degC*			**Min Temp.** *degC*
200	180	20	24	85	5	5	2	15	----

As clarified by Table 4, two context elements have conflicts, Precipitation and Max Temp. One context element is not available, the Min Temp.

Table 5 clarifies the context data for the object level.

Table 5 The Context Objects Values

Rainfall	**Soil Saturation**	**Wind**	**Temperature**
Heavy Rainy	Severe Soil Saturation	Wind Storm	Low

The context data for situation level is clarified by Table 6 as all previous values for context object indicate a potential flood.

Table 6 Context Situation Value

Flooding
Potential Flood

V. RESULTS AND EVALUATION

This Section articulates the experiments, results and evaluation for our proposedframework.We start by evaluating IPQP as a separate part to be ensure that it works well alone before testing the whole framework.

As mentioned earlier in Section, IPQP addresses context conflicts, missing values, and context erroneous values. There are many experiments areconducted using Southampton weather dataset and Cardiff dataset that are described in Section; the following results shows just one experiment with Southampton weather dataset.

A. Evaluating IPQP

For context conflicts resolving, IPQP predicts the valid values among set of context conflicted values, and then chooses the best value according to quality policy. The results shown according to IPQP steps.

The results explain the results of two examples; the first one is for applying IPQP to resolve context conflicts and the second is for applying IPQP to handle context erroneous and missing values.

Table 7 shows a set of conflicted context values, which are the values for Max Temperature (Tmax) which is a variable in Southampton dataset.Table 7 showsthe results of the first step of IPQP which is applying the algorithm RCCAR to exclude the invalid values. Invalid values are the values that have lower affirmation values and values that are so far from the value with the greatest affirmation based on the permitted value(in our case, $\Delta(\boldsymbol{Tmax}) = \mathbf{2}$).

Table 7 Results of IPQP Algorithm for Resolving Context Conflicts- the First Step

Tmax Conflicted Values	Total_Affirmation using RCCAR %	Valid/ Invalid	Pass to the Next Step	Notes
9	55	valid	pass	the best value
30	30	invalid	excluded	
10.4	53	valid	pass	in the accepted range $\Delta(Tmax)$
2	0.21	invalid	excluded	

Then, IPQP computes the quality parameters for each valid value based on the quality policy.Based on quality attributes for each valid value, IPQP produce the best ideal solution and compute the distance between each valid value and the ideal solution using TOPSIS, then select the best value accordingly. Figure 9 shows the result for this last step.

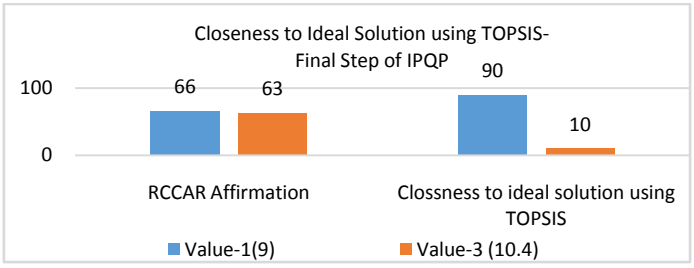


Figure 9 Results of IPQP for Resolving Context Conflicts -The Final Step of IPQP

The coming example presents the situation of IPQP algorithm performance for handling context missing values and erroneous values. In this situation, the same solution is applied. IPQP predicts the best valid value based on all values in previous context history database.As shown by Figure 10, IPQP provided a good success in prediction of the correct value with a big distance of other values even with the worst case (for 5 years) as the correct value has an affirmation value exceeds 50% and a distance of the nearest wrong value (at least 28.57%) which is a good distance value.

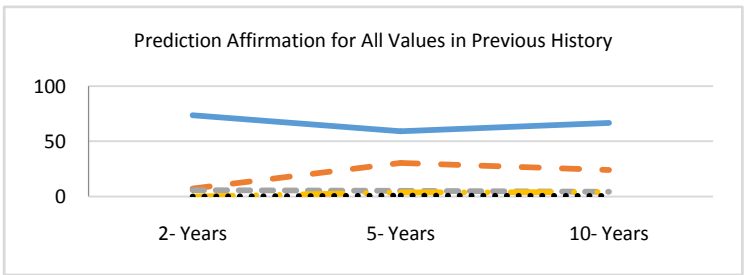


Figure 10 Results of IPQP in Handling Context Missing Value and Erroneous Values

The main drawback here would be if there were many context elements that have erroneous values. This case will lead to low affirmation for most context elements as the affirmation for each context element depends on other context elements values that should be valid values. This point needs more experiments to study.

B. Framework Evaluation

To evaluate our framework MCFQoC, a case study for floods forecasting is used. Flood forecasting system is considered as a context-aware system that works to monitor and observe changes in natural areas that are prone to flooding and working on analysis and then infer the incidence of possible flooding according to the context that has been captured. The framework works to allow the system users to identify and establish the qualitypolicyfor context and then control the quality of context automatically using IPQP and produce different quality indicators that enable the system to make decisions accordingly.

QualityPolicy Setting

Before applying IPQP and producing the QoC indicators, the qualitypolicy for each level will be determined in terms of qualityparameters and their relative weights.The user should determine the desired qualityparameters in each contextlevel according to the application area and the problem nature. For our case study in hand, the quality parameters are shown in Table 8.

Table 8 The Admitted QualityParameters According to the Application User

The Admitted QualityParameters for **ContextElements**		
Reliability	Timeliness	Accuracy
The Admitted Qualityparameters for **ContextObjects**		
Credibility	Usability	
The Admitted Qualityparameters for **ContextSituation**		
Credibility		

According to our framework, the first step is achieving a pairwise comparison for all qualityparameters in each context level. As shown by Table 9, application'suser should perform a pairwise comparison using the scale [0..9].

Table 9 Achieving the Pairwise Comparison between the QualityParameters

	Reliability	**Timeliness**	**Accuracy**
Reliability	1/1	3/1	7/1
Timeliness	1/3	1/1	5/1
Accuracy	1/7	1/5	1/1

After Applying AHP as a part of our framework, weights of qualityparameters are produced as shown in Table 10.

Table 10 Weights of QualityParameters for ContextObjects

Parameter	**Weights**
Reliability	0.65
Timeliness	0.28
Accuracy	0.07

According to the qualitypolicy for contextobject, there areonly two parameters thus no need to perform pairwise comparison. The user simply determines the weights according to the application area as shown by Table 11.

Table 11 The ContextobjectQualityParameters Weights

Parameter	**Weights**
Securitylevel	0.50
Credibility	0.50

For contextsituation,based on quality parameters model, there is only one parameter *Credibility* with 100% relative weight.

Results of Framework Evaluation

This section introduces the results of MCFQoC framework evaluation starting with results of IPQP as a crucial part.

As shown by Table 4, we addressed two shortcomings: context conflicts for two context element (Precipitation and MaxTemp. context elements) and a missing value (MinTemp context element).

Based on IPQP, the affirmation for all context elements was good except for saturation_level context element, which was 0.05% as shown by Table 12. This low affirmation indicates that the value of this context element could be erroneous value.

Table 12 Results of Affirmation for All Context Elements to Detect the Potential Erroneous Values

Rainfall Precipitation	Rainfall Period	Wind Speed	Saturation Level	Max Temp.
0.63	0.63	0.80	0.05	0.83

In this stage, context conflict for precipitation and max Tempis resolved using RCCAR algorithm and TOPSIS method.

For precipitation, the results show that two selected values will pass to the next step of IPQP algorithm. These two values are 200 and 180 as shown by Table 13.

Table 13 Results of RCCAR for Precipitation Context ElementIPQP First Step

Conflicted Values	Prediction Affirmation (Probability of Correctness) using RCCAR	Valid/ Invalid	Pass to the Next Step	Notes
200	4.15	valid	pass	the best value
180	3.50	valid	pass	
20	0.15	invalid	excluded	

Similar to Precipitation, Table 14 shows the output of RCCAR algorithm for MaxTemp. context elements. The two selected values 5 and 2 will pass to the next step of IPQP algorithm.

Table 14 Results of RCCAR for Max Temp. Context Element IPQP First Step

Conflicted Values	Prediction Affirmation (Probability of Correctness) using RCCAR	Valid/ Invalid	Pass to the Next Step	Notes
5	4.70	valid	pass	the best value
2	3.15	valid	pass	
15	0.02	invalid	excluded	

Now, based on the relative weights, which are produced using AHP, TOPSIS method is used to produce the best value, which is closest to the ideal solution as shown by Table 15. and Table 16 for precipitation and max temp. respectively.

Table15 The Relative Closeness to the Ideal Solution for Precipitation Context Element

	Closeness to the Ideal Solution
200	0.83
180	0.17

Thus, the best value is 200 with 0.83 closeness value to the ideal solution.

Table 16 The Relative Closeness Ratio from the Ideal Solution for Max Temp. Context Element

	Closeness to the Ideal Solution
5	0.04
2	0.96

Therefore, the best value is 2 with 0.96 relative closeness ratio to the ideal solution.

As shown by Table 12. probability of correctness (affirmation) using RCCAR for all context elements were good except for saturation_level context element, which was 0.05%. The low affirmation for

saturation_level indicates that the value of this context element could be erroneous value.By applying IPQP algorithm for this situation, the result was as shown by Table 17.

Table 17 Results of IPQP for Handling Erroneous Values for Saturation_Level

No.	Predicted Values For Saturation_level	Affirmation (%)	Notes
1	87%	72%	Selected Value
2	66%	65%	
3	95%	60%	
4	60%	50%	

As shown by Table 17, the predicted value is so far from the old value with 87%. The new predicted value seems more reasonable for flooding situation.

As shown by Table 4, one context element has a missing value; this context element is Min Temp. By applying IPQP, the missing value is predicted using RCCAR. The results are shown by Table 18.

Table 18 Results of IPQP for Handling Missing Values for Min Temp. Context Element

No.	Predicted Values	Affirmation (%)	Notes
1	-1	73%	Selected Value
2	0	66%	
3	-0.5	60%	

Producing the Quality Parameters/Indicators Using the Proposed Framework

After applying IPQP, the different quality parameters and indicators are produced using the quantification methods that have been introduced in our framework. Table 19 shows the different quality parameters and total quality for each piece of context. In addition, it shows results for the general indicators beside the quality parameters and total quality for different levels.

Table 19 Results ofMCFQOCFramework for Different Context Levels

Qualitypolicy for Contextsituation Level		**Parameter: Credibility Weight: 100%**			
quality degree	0.68	**Flooding**			
average of the credibility parameter	0.68	Total Quality 0.68			
The average for the non-validity parameters	-------	Credibility 0.68			
The average for all qualityparameters	Credibility=0.68				
Qualitypolicy for Contextobject Level		**Parameter: Credibility Weight: 50%** **Parameter: Securitylevel Weight: 50%**			
quality degree	0.79	**Rainfall** Relative Weight **(30%)**	**Wind** Relative Weight **(20%)**	**Soil Saturation** Relative Weight **(30%)**	**Temperature** Relative Weight **(20%)**
average of the credibility parameter	0.67	Total Quality 0.85	Total Quality 0.64	Total Quality 0.86	Total Quality 0.82
The average for the non-validity	0.92	Credibility 0.70	Credibility 0.61	Credibility 0.72	Credibility 0.64

parameters						
The average for all quality parameters	Credibility= 0.67 Security-level= 0.92	SecurityLevel 1.00	Securitylevel 0.66	Securitylevel 1.00	SecurityLevel 1.00	
Qualitypolicy for Context Elements Level		**Parameter: Reliability Weight: 0.65%** **Parameter: Timeliness Weight: 0.28%** **Parameter: Accuracy Weight: 0.07%**				
quality degree	0.66	**Precipitation** Relative Weight **(60%)**	**Period** Relative Weight **(40%)**	**Wind Speed** Relative Weight **(100%)**	**Saturation Level** Relative Weight **(100%)**	**Max Temp.** Relative Weight **(50%)**	**Min Temp.** Relative Weight **(50%)**
average of the credibility parameter	0.67	Total Quality 0.77	Total Quality 0.75	Total Quality 0.37	Total Quality 0.71	Total Quality 0.69	Total-quality --------
The average for the non-validity parameters	0.42 (Accuracy)	Reliability 0.80	Reliability 0.80	Reliability 0.32	Reliability 0.95	Reliability 0.53	Reliability ------
		Timeliness 0.75	Timeliness 0.75	Timeliness 0.50	Timeliness 0.25	Timeliness 1.00	Timeliness --------
		Accuracy 0.50	Accuracy 0.33	Accuracy 0.33	Accuracy 0.33	Accuracy 1.00	Accuracy -------
The average for all qualityparameters	Reliability= 0.57 Timeliness= 0.54 Accuracy= 0.42	Probabilityof Correctness 0.63	Probabilityofcorrectness 0.63	Probabilityofcorrectness 0.80	Probabilityofcorrectness 0.83	Probabilityofcorrectness 0.83	Probabilityofcorrectness 0.73
		Credibility 0.70	Credibility 0.70	Credibility 0.61	Credibility 0.72	Credibility 0.80	Credibility 0.73

The highlighted results are those results handled by IPQP for erroneous and missing values. Credibility for MinTemp.is produced using only probability of correctness as it is a missing value and so there is no metadata available to compute its reliability and timeliness.
The results produced using our framework indicate many different aspects. The main important results are the quality parameters for each pieceofcontext in different context levels according to the qualitypolicy.The results of the framework are as described below:

1. The Different QualityParameters for each PieceofContext

Table 19shows the results of all qualityparameters for each pieceofcontext. The results contain the values for qualityparameters that are assigned in the qualitypolicy and other qualityparameters that are used to calculate other parameters such as probability_of_correctness and credibility in contextelements level. The results show that:

- The qualityparameters in higher levels are affected by the related parameters in lower level for the synthesis parameters, which are calculated based on the parameters values in the lower level such as credibility for contextsituation and credibility for composite objects. For example, the qualityparameters values for Rainfall object and the qualityparameters values for the contextelements that compose the Rainfall object; these context elements are Precipitation and Rainfall Period. We can see that the credibility of the object isthe total credibility of the contextelements whereas the security_level of the object is equal to 1; it was not affected by the parameter in the low level because it is not a synthesis parameter.Another example is Temperature contextobject. For this contextobject, only one contextelement was available, that is Max Temp where Min Temp is then predicted with 0.47% credibility value which is low because it is affected by the absence of reliability and timeliness for this context element. This low credibility value for context element has affected the object credibility with a relative weight of 50%. Finally, as shown by Figure 11, credibility for Flooding contextsituation is affected by boththe credibility and the weights of contextobjects, which compose the situation.

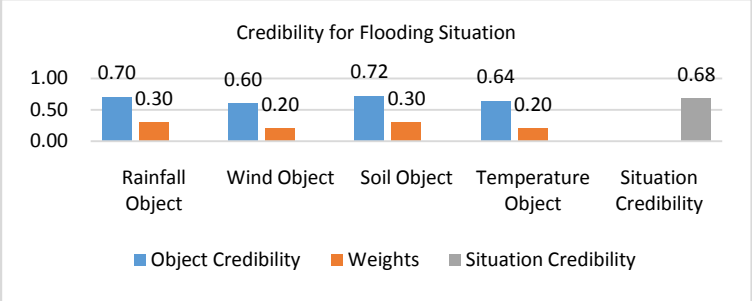


Figure 11 Credibility for Flooding Situation

- The average value of credibility for each level does not make a sense for the credibility in the upper level (see Figure 12). For example, the average of credibility in contextobjects level is less than credibility in contextsituation level. The average of credibility in each level is a general indicator that indicatesthe need for attention to this parameter in that certain level; it draws attention to the underlying causes of this weakness, if any.

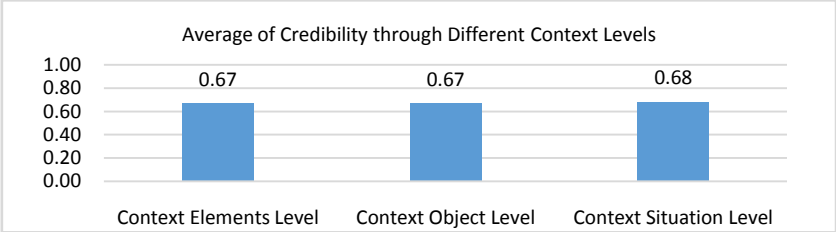


Figure 12 Average of Credibility through Different Context Levels

- As shown by Figure 13, for different contextelements, there is a contrast in the result of reliability parameter that is based on sensors and the probability_of_correctness that is based on the context history. This result indicates the importance of probability_of_correctness as a validity quality indictor beside the reliabilitywhich based on the sensor efficiency that prone to breakdowns for different physical reasons.

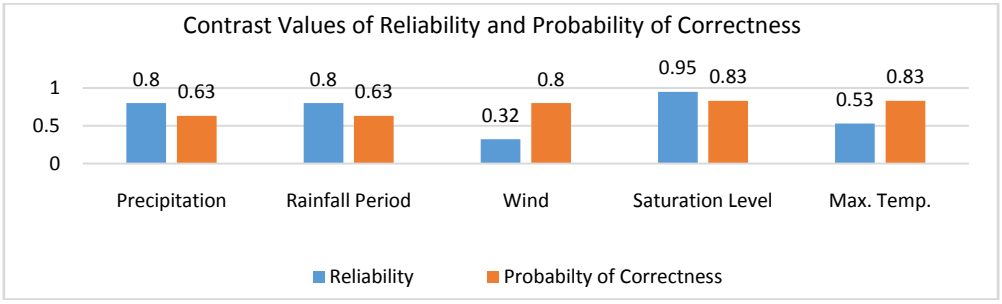


Figure 13 Contrast Values of Reliability and Probability_of_Correctness

- Credibility reflects the validity of context. It makes balance between the parameters that affect validity, which are reliability, timeliness, and probability_of_correctness. As shown by Figure 14, reliability for such contextelement could be low (0.53) whereas the probability_of_correctness is high (0.83) as it is for Max Temp contextelement. The low value of reliability is due to a fault with the sensor that senses this contextelement. The resulted credibility is more accurate with (0.80) based on the high value of probability_of_correctness parameter.

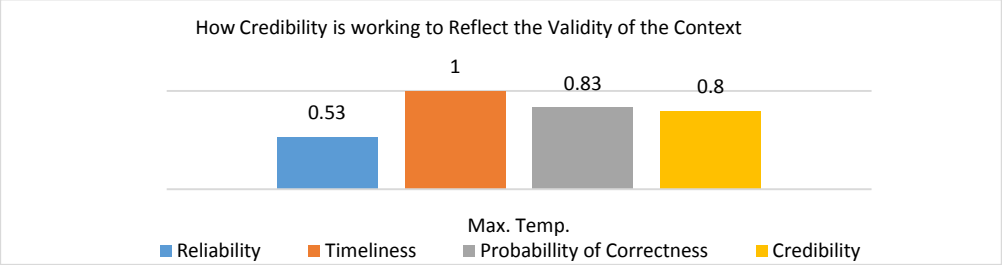


Figure 14 How Credibility Works to Reflect the Validity of the Context

2. The Total_Quality for each PieceofContext According to the QualityPolicy

As shown by Table 19, the total value of quality for each pieceofcontext is affected by two factors: parameters weights in qualitypolicy and parameters values for that pieceofcontext.

The totalquality for each pieceofcontextdoes not necessary reflect the validity. As shown by Figure, for all contextelements, there are differences between credibility value, which reflects the validity and the value of quality according to qualitypolicy. This is the reason that ourframework calculates the credibility for each pieceofcontext even it is not a part of the adopted qualitypolicy.Qualitypolicy reflectsquality requirements, which could be perfection aspects such as representationconsistency, however credibility should come first, and both of them are important for the system.

3. The Average of the Credibility Parameter for each Level

As shown by Table 19, this indicator will reflect the validity in general for each level regardless the different piecesofcontext. Although we should think about credibility for each pieceofcontext separately because of the relative weights of elements that compose the context; this indicator is still important because it makes the CAS's user and developer thinking to raise up the performance of the part of system that responsible for producing the context in the levelthat needs that.

4. The Average of the Non-Validity Parameters for each Context-Level

This general indicator helps the CAS's user focusing on non-validity parameters in the qualitypolicyapart from the credibility, and thinking about the conditions that affect their values. These parameters are security and representation_consistency parameters. This will help improving imperfection of the context. The total_quality of context level does not indicate the problems that are related to these perfection parameters as we can find high credibility and low non-validity parameters whereas the average could be fine as shown by Table 19.

5. The Average for Different QualityParameters in each ContextLevel

To allow the CAS's user focusing on each parameter that contributes in qualitypolicy for each context level; the average of each parameter in the level should be produced. The system user should concentrate on the parameters that hold the higher importance level according to qualitypolicy. As Shown by Table, the reliability parameter should receive the most attention to improve its value as it forms the majority of the quality value even that the accuracy level is low, but it does not matter because the accuracy weight is so low.

6. Results without Handling Missing and Erroneous Context Values

This section shows the main results in case that the context missing values and erroneous values havenot been handled using IPQP. As shown by and Table20 and Figure 15, the results show how much IPQP improves the context credibility.

Table 20 Results withoutHandling Missing and Erroneous Context Values

Quality-policy for Context-situation Level		**Parameter: Credibility** **Weight: 100%**			
average of the credibility parameter	0.51	**Flooding**			
		Credibility 0.51			
Quality-policy for Context-object Level		**Parameter: Credibility** **Weight: 50%** **Parameter: Security-level** **Weight: 50%**			
average of the credibility parameter	0.51	**Rainfall** Relative Weight **(30%)**	**Wind** Relative Weight **(20%)**	**Soil Saturation** Relative Weight **(30%)**	**Temperature** Relative Weight **(20%)**

		Credibility 0.70	Credibility 0.61	Credibility 0.33	Credibility 0.40		
Quality-policy for Context-elements Level		**Parameter: Reliability Weight: 0.65%** **Parameter: Timeliness Weight: 0.28%** **Parameter: Accuracy Weight: 0.07%**					
average of the credibility parameter	0.52	**Precipitation** Relative Weight **(60%)**	**Period** Relative Weight **(40%)**	**Wind Speed** Relative Weight **(100%)**	**Saturation Level** Relative Weight **(100%)**	**Max Temp.** Relative Weight **(50%)**	**Min Temp.** Relative Weight **(50%)**
		Probability of Correctness 0.63	Probability-ofcorrectness 0.63	Probability-ofcorrectness 0.80	Probability-ofcorrectness 0.05	Probability of correctness 0.83	Probabilityofcorrectness -----
		Credibility 0.70	Credibility 0.70	Credibility 0.61	Credibility 0.33	Credibility 0.80	Credibility -----

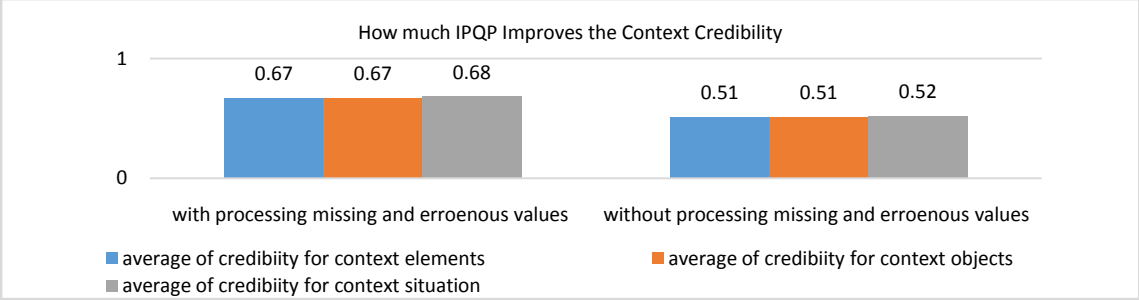


Figure 15 How much IPQP Improves the Context Credibility

VI. CONCLUSIONS AND FUTURE WORK

This research introduces a novel comprehensive framework for QoC. According to results, the proposed framework has achieved good success in assuring the QoC in CASs. It combines theoretical components with a system of QoC assurance. Theoretical components include a new definition for QoC and a new model for QoC parameters. The system of QoC assurance contains a method for determining quality policy, qualitycontrol processes for handling contextconflicts, contextmissing values, and context erroneous values. The quality control processes aim in general at supporting context uncertainty. Theoretical parts of the framework are vital as they inspired us to develop other parts starting with parameters quantification methods and ending with qualitycontrol processes. These theoretical parts should be essential parts for any future frameworks.The framework established based on the idea of context multilayers. This idea helps producing a more accurate QoC parameters model, and also more accurate quantification methods for these parameters.Using AHP method for assigning the weights of QoC parameters within the quality policyis a new idea. Exploiting AHP to determine qualitypolicy has produced a more accurate qualitypolicy as it examines the consistency of parameters weights before adopting the policy.IPQP approach is introduced as a quality control process within the framework. It is an extension to RCCAR whichisintroduced asa sensed context quality control process [52]. It handles three main sensed context shortcomings: context conflicts, context missing values, and context erroneous values. IPQP has achieved a significant improvement for context quality.Credibility parameter, which is a new important contribution in this research, has achieved a remarkable success in terms of resolving context uncertainty as it produced a strong indication for context validity. Moreover, credibility significantly minimized the errors that could be produced by other parameters such as reliability and timeliness.As shown by experiments, the most important QoC parameters for resolving context uncertainty were the validity parameters such as reliability, timeliness, and credibility.The qualitylevel of the context in the lowerlayers affects the qualitylevel of the context in the upperlayers just with the validity parameters, which have a synthesis nature such as completeness, reliability, timeliness, and credibility. Therefore, improving these parameters values in lowerlevel affected significantly the QoC in upper level.

On the other hand, studying the success of framework in supporting context reasoningand context uncertainty resolving in real CAS environment could be better in order to assess the success of the

framework accurately. In addition, this research did not conduct complete context management.In future work we will aim to develop a complete context management framework centered on the QoC framework proposed in this paper.

REFERENCES

[1] B.Thomas and M. Schiffers, "QoC: What It Is And Why We Need It", In Proceedings of the 10th Workshop of the Open View University Association: OVUA'03 , 2003.

[2] A. Manzoor, H. Truong and S. Dustdar, "QoC: Models and Applications for CAS in Pervasive Environments", Special issue on Web and Mobile Information Services, 2010.

[3] A. Manzoor, H. Truong and S. Dustdar,"On the Evaluation of QoC", Proceedings of the 3rd European Conference on Smart Sensing and Context, Springer-Verlag Berlin, Heidelberg, 2008.

[4] Y. Bu, T. Gu, X. Tao, J. Li, S. Chen andJ. Lu, "Managing QoC in Pervasive Computing", Sixth International Conference on Quality Software (QSIC'06), 2006.

[5] K. Henricksen and J. Indulska, "Modeling and using imperfect context information", Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, IEEE, 2004.

[6] M. Krause and I. Hochstatter, "Challenges in Modelling and Using Quality of Context (QoC)", T. Magedanz et al.(Eds.): MATA 2005, LNCS 3744, pp. 324–333, Springer-Verlag Berlin Heidelberg, 2005.

[7] J.B. Filho,A.D. Miron, andI. Satoh, "Modeling and Measuring Quality of Context Information in Pervasive Environments", 2010 24th IEEE International Conference on Advanced Information Networking and Applications, 2010.

[8] H. Hegering, "Management Challenges of Context-Aware Services in Ubiquitous Environments", DSOM 2003, LNCS 2867, IFIP International Federation of Information Processing, 2003.

[9] J.B. Filho andN. Agoulmine,"A Quality-AwareApproach for Resolving Context Conflicts in Context-Aware Systems", In proceeding of IEEE/IFIP 9th International Conference on Embedded and Ubiquitous Computing, EUC 2011, Melbourne, Australia, 2011.

[10] A. Toninelli, A. Corradi and R. Montanari, "A Quality of Context-Aware Approach to Access Control in Pervasive Environments", The Second International ICST Conference on Mobile Wireless Middleware, Operating Systems, and Applications, Berlin, 28 April 2009.

[11] F. Paganelli, G. Bianchi and D. Giuli, "A Context Model for Context-aware System Design towards the Ambient Intelligence Vision: Experiences in the eTourism Domain", Universal Access in Ambient Intelligence Environments 2006.

[12] N. Malik, U. Mahmud and Y. Javed, "FutureChallenge'sinContext-Aware Computing", IADIS International Conference, 2007.

[13] H. Lee, J.S. Choi and R. Elmasri, "A Classification and Modeling of the Quality of Contextual Information in Smart Spaces", PERCOM 2009, IEEE International Conference, 2009.

[14] Z. Abid andS. Chabridon, "A fine-grain Approach for Evaluating the Quality of Context", Pervasive Computing and Communication Workshop PERCOM 2011, IEEE, 2011.

[15] A. Manzoor, "Quality Aware Context Information Aggregation System for Pervasive Environments", WAINA 2009, IEEE Conference, 2009.

[16] E. Kim, "A Context Management System for Supporting Context-Aware Applications", EUC 2008, IEEE Conference, 2008.

[17] M. A. Razzaque,Simon Dobson and Paddy Nixon, "Categorization and Modeling of Quality in Context Information", CiteseerX, 2006.

[18] T. L.Saaty, "Relative Measurement and its Generalization in Decision Making: Why Pairwise Comparisons are Central in Mathematics for the Measurement of Intangible Factors - The Analytic Hierarchy/Network Process", RACSAM (Review of the Royal Spanish Academy of Sciences, Series A, Mathematics) 102 (2): 251–318, 2008.

[19] http://www.southamptonweather.co.uk/sotonhist.php(Nov. 5, 2012).

[20] I. Witten and E. Frank, "Data Mining: Practical machine learning tools and techniques", Morgan Kaufmann, 2nd Edition, 2005.

[21] http://www.metoffice.gov.uk/public/weather/forecast/cardiff(Nov. 5, 2012).

[22] ISO 9000:2005 Quality Management Systems - Fundamentals and Vocabulary, ISO British Standard Institution, 2005.

[23] M.BäuyÄukyazand M. Sacu, "The Analytical Hierarchy and Analytic Network Processes", Hacettepe Journal of Mathematics and Statistics Volume 32, 2003.

[24] T. L. Saaty, "How to make a decision: The Analytical Hierarchy Process", European Journal of Operational Research 48(1990) 9- 26, North Holland, 1990.

[25] T. L. Saatyand K. Peniwati, “Group Decision Making: Drawing out and Reconciling Differences”. Pittsburgh, Pennsylvania: RWS Publications. ISBN 978-1-888603-08-8, 2008
[26] C. L. Hwang andK. Yoon,"Multiple Attribute Decision Making: Methods and Applications", New York: Springer-Verlag, 1981.
[27] K.Yoon, "A reconciliation among discrete compromise situations". Journal of Operational Research Society 38. pp. 277–286., 1987.
[28] C. Hwang andK. Yoon, "A new approach for multiple objective decision making". Computers and Operational Research 20: 889–899, 1993.
[29] K. Yoon andC. Hwang, "Multiple Attribute Decision Making: An Introduction", SAGE publications, 1995.
[30] N. Brgulja, R. Kusber, and K. David, "Validating Context Information in Context Aware Systems", Sixth International Conference on Intelligent Environments, 2010.
[31] K. Sheikh, M. Wegdam and M.V. Sinderen, "Middleware Support for Quality of Context in Pervasive Context-Aware Systems" ,Proceedings of the Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops(PerComW'07), 2007.
[32] S. McKeever, J. Ye, L. Coyle and S. Dobson, "A multilayered uncertainty model for context aware systems", The international conference on Pervasive Computing, 2008.
[33] J. Ye, S. McKeever, L. Coyle, S. Neely and S. Dobson,"Resolving Uncertainty in Context integration and Abstraction", ICPS’08, ACM, 2008.
[34] S. McKeever, J. Ye, L. Coyle, and S. Dobson, "A Context Quality Model to Support Transparent Reasoning with Uncertain Context", Dublin Institute of Technology, 2009.
[35] K. Henricksen, *J.* Indulska, and A. Rakotonirainy,"Modelling Context Information in Pervasive Computing Systems", Springer-Verlag Berlin Heidelberg 2002, Pervasive 2002, LNCS 2414, pp. 167–180, 2002.
[36] S. Poslad, "Ubiquitous Computing: Smart Devices, Environments and Interactions", John Wiley & Sons Ltd, 2009.
[37] M. Baldauf, S. Dustdar and F. Rosenberg, "A survey of Context-aware Systems", Int. J. Ad Hoc and Ubiquitous Computing, Vol. 2, No. 4, 2007.
[38] http://www.cimms.ou.edu/~doswell/barcelona/flashf.html[Nov. 21, 2013].
[39] https://www.gov.uk/check-if-youre-at-risk-of-flooding[Nov. 21, 2013].
[40] C. Anderson, “What is a Quality Management System”, Bizmanualz, Nov 18, 2013.
[41] Science Park Association, “Good Practice Guide ,Electronic Quality Management System for SMEs”, United Kingdom, Compliance Control Ltd, 2014.
[42] D. Hoyle, “ISO 9000 quality systems handbook”, ISBN 0-7506-6785-0, 2005
[43] H. Kerzner, “Project Management: A Systems Approach to Planning, Scheduling, and Controlling”, John Wiley, ISBN 978-0-470-50383-6, 2009.
[44] J. Pike and R. Barnes, “TQM in action: a practical approach to continuous performance improvement”, ISBN 0-412-71530-9, 1998.
[45] K. R. Grimes, “ISO 9001:2000: a practical quality manual”, ISBN 0-87389-555-X, 2002.
[46] G. D. Abowd, G. R. Hayes, G. Iachello, J. A. Kientz, S. N. Patel, M. M. Stevens andK. N. Truong, “Prototypes and Paratypes: Designing Mobile and Ubiquitous Computing Applications”, PERVASIVE computing, IEEE CS and IEEE ComSoc, 2005.
[47] M. Lethrech, I. Elmagrouni, M. Nassar and A. Kriouile, “Domain Specific Modeling Approach for Context-aware Service Oriented Systems”, 2014 International Conference on Multimedia Computing and Systems (ICMCS), IEEE Proceedings, Marrakech, 14-16 April 2014.
[48] K. Mitra, A. Zaslavsky and C. Ahlund, “Context-Aware QoE Modelling, Measurement, and Prediction in Mobile Computing Systems”, IEEE Transactions on Mobile Computing, Vol. 14, Issue 5, pages 920-936, May 2015.
[49] Y. Lu, M. Motani and W. C. Wong, “A QoE-Aware Resource Distribution Framework Incentivizing Context Sharing and Moderate Competition”, Transactions on Networking, IEEE/ACM, Volume: PP, Issue: 99, pages 1-14, March 2015.
[50] J. Berri, “Context reasoning for mobile systems”, 2014 World Congress on Computer Applications and Information Systems (WCCAIS),IEEE Proceedings, 2014.
[51] B.Mortazavi,M.Pourhomayoun,H.Ghasemzadeh,, R.Jafari, C.K.Roberts and M.Sarrafzadeh, , “Context-Aware Data Processing to Enhance Quality of Measurements in Wireless Health Systems: An Application to MET Calculation of Exergaming Actions “,Internet of Things Journal, IEEE, Volume 2, Issue 1, pages: 84 – 93, 2015.
[52] A. A. Al-Shargabi and F. Siewe, "Resolving Context Conflicts Using Association Rules (RCCAR) to Improve Quality of Context-Aware Systems", ICCSE 2013, IEEE conference, 2013.