Towards Measurable Efficient and Effective Metrics for Quality and Cost of Context

Vers des métriques efficace et effective mesurables pour la Qualité et le Coût du Contexte

Kanaka Sai Jagarlamudi, Arkady Zaslavsky, Seng W. Loke, Alireza Hassani, Alexey Medvedev

School of Information Technology, Deakin University, Melbourne, Australia
{kjagarlamudi, arkady.zaslavsky, seng.loke, ali.hassani, alexey.medvedev}@deakin.edu.au

ABSTRACT. Despite the potential benefits of the context-driven intelligence delivered by Context Management Platforms (CMP), the lack of efficient and effective metrics for measuring Quality and Cost of Context (QoC and CoC) hinders them from uptake and commercialisation. Furthermore, the CMPs might have limited abilities to incorporate efficient QoC drivers and a suboptimal selection of QoC-aware context providers. This paper proposes QoC and CoC metrics and introduces a conceptual architecture to achieve the QoC and CoC awareness in CMPs, aiming to improve their efficiency and consumer experience.

RÉSUMÉ. En dépit des bénéfices potentiels d'une intelligence guidée par le context que délivrent les plates-formes de management des contexts (CMP), le manqué de métriques fonctionnelles et efficaces pour mesurer la qualité et le coût du contexte (QoC and CoC) limitent l'adoption et la commercialisation de ces métriques. En outre, ces CMPs peuvent avoir des capacités limitées pour incorporer une QoC efficace et une sélection sous-optimale des fournisseurs de contextes sensibles à la QoC. Ce papier pose des métriques qoC et CoC, introduit une architecture conceptuelle pour réaliser la sensibilité à QoC et CoC dans les CMPs, visant à améliorer leur efficacité et l'expérience du consommateur de contextes.

KEYWORDS. Quality of Context, Cost of Context, QoC-awareness, CoC-awareness, Context Management Platforms.

MOTS-CLÉS. Qualité du Contexte, Coût du Contexte, sensibilité à QoC et CoC, plates-formes de management du contexte.

1. Introduction

The context research community has been aware of Quality of Context (QoC) 's significance for the past two decades. It was initially defined in [1] as "a*ny information that describes the quality of information that is used as context information"*. In context-aware ecosystems, the context providers' (sources such as sensors) responses may contain individual elements known as low-level context. We define them as *contextual elements*; these are further inferred/aggregated by CMP to produce the high-level contextual information delivered to the context consumers (recipients). Generally, the metrics of *QoC Indicators* (context quality properties) are used to measure and validate the provided contextual elements' ability to satisfy the context consumers' requests; we define the required QoC indicators bound to each contextual element as *Quality of contextual elements (QoC elements)*. The statement made in work [2] establishes the importance of providing the valid context, *"context acts like a set of constraints that influence the behaviour of a system embedded in a given task"*. So, its validity is crucial for a context consumer to make the appropriate decisions. Thus, enabling CMPs with QoC related features promotes valid context provisioning, increasing their context consumers' experience. In contrast, due to the lack of standard definitions related to Cost of Context (CoC), this paper defines it as "*a parameter representing the CMP's costs for context processing and acquisition from context providers. It translates into the monetary cost to context consumers*". Thus, reducing the context acquisition costs by selecting low-cost providers based on CoC awareness allows CMPs to offer context for lower costs, therefore increasing the overall profitability for the ecosystem.

The following use case explicitly defines the above terms and significance of this research. Provision of related information by contextual intelligence concerning fire alerts enables an effective and proactive response from concerned individuals/authorities. The entities on fire alerts include the places (e.g., buildings) or areas (e.g., forests). The contextual elements produced from these entities may include the potential fire threats, the current rate of fires, the number of people to be rescued, humidity and wind conditions, and nearby police and ambulances. Each contextual element's validity and cost depend on the values of associated QoC elements (e.g., freshness, accuracy, representation) and their CoC. In contrast, on the one hand, the context consumers could be the concerned authorities (e.g., fire control, police and ambulances); for effective responses, they require contextual elements with high QoC regardless of the CoC (monetary cost imposed by CMP). On the other hand, the context consumers could be the trip planner applications: notifies the civilians of potential fire threats and support services at the destinations; these applications can be satisfied with relatively less QoC than authorities.

For increasing the performance efficiency, user experience and profitability, the CMP must select the providers that satisfy both consumers above' individual QoC requirements with the lowest acquisition costs. It is achieved by enabling the following inter-related features in the CMP: QoC and CoC-aware context providers' selection, QoC computation and QoC validation. QoC and CoC-aware selection select context providers by projecting their QoC and CoC outcomes. Hence, QoC-aware selection increases the CMPs' performance efficiency by reducing the need for providers' reselection due to invalid QoC provision. It also increases context consumers' experience by fulfilling individual QoC requirements. The CoC-aware selection increases the profitability by selecting the context provider with low costs. Then, QoC computation measures the QoC elements' values in each acquired contextual element. Based on these values, the QoC validation determines the context elements' validity towards serving the request, thereby ensuring only a valid context is delivered to the consumer; therefore, increasing their experience.

This paper aims to present a framework called conCQeng (context Cost-Quality engine) that potentially enables the features above in CMPs upon integration. Also, we define the novel inter-related approaches called Relative Reputation (RR) and Assurance to perform the personalised QoC and CoC- aware selection. The organisation of this paper is as follows: Section 2 provides the related work, Section 3 presents our system architecture, and Section 4 provides the conclusion and future work.

2. Related work

Most of the advanced CMPs including, Context-as-a-Service (CoaaS) [3] (the focused CMP for our framework), possesses limitations in enabling QoC and CoC features; this section provides key related work to address these limitations. The QoC modelling techniques are used to measure the QoC elements. The work in [4] presented a modelling technique with a wide range of QoC indicators and their computation formulas. Therefore, we adopt their approach to perform QoC computation.

As given in work [5], Service Level Agreements (SLAs) contain the quality guarantees that include QoC. These guarantees can act as input for QoC-aware selection; thus, it is essential to form SLAs between the context providers and CMP. Besides, they also contribute to profitability in CoC returns, paid as penalties for the context providers' QoC violations. Contrarily, SLA also contains the context providers' CoC details; hence, contributing to CoC-aware selection. Therefore, we form the SLA between the context providers and CMP and CMP and context consumer, containing the details of QoC guarantees, CoC, and QoC violation penalties. The reflection of the SLA between the CMP and context consumer is out of scope.

The QoC guarantees of context providers are defined based on their design-time capabilities. However, these capabilities degrade over time due to external and internal inconsistencies, thus achieving inaccurate QoC-aware selection results. For addressing these challenges, it is best to perform the QoC-aware selection based on the run-time evidence on QoC capabilities. The reputation models can indicate such evidence by providing the context providers' satisfaction rate of their given SLAs. The work in [6] performs the reputation based QoC-aware selection. One of their approaches is based on Social IoTs (SIoTs), where consumers locally store the reputation of the context providers and share it with the service-seeking consumers. However, implementing this model is functionally challenging in modern-day CMPs as they abstract the providers and consumers. In the second approach, reputation is assessed and stored at the CMP level; this assessment is performed generally based on the QoC provision to diverse consumers. In contrast, acceptancy towards provided QoC varies between the consumers. Thus, requiring a personalised reputation assessment of context providers for each consumer.

The work in [6] discussed the modelling of potential reputation models in CMPs. One of these models aimed to perform the personalised QoC aware selection. In this model, the reputation is assessed using consumers' feedback on the QoC and their end-users personal preferences. However, CMPs lack the privilege to access end-users personal data. Contrarily, entirely relying on feedback to assess the satisfaction rate may attain inaccurate results. We address these challenges by performing a personalised reputation assessment of context providers with each context consumer, omitting the need for end-user data. Furthermore, our model's reputation is assessed based on context validity – computed using the metrics and context consumers' reports on QoC violations. Thus, effectively assessing the reputation while providing a degree of personalisation.

3. System architecture

As depicted in Figure 1, on the one hand, concerned authorities and Trip planning individuals are context consumers. On the other hand, sensors associated with fire alert entities generating the contextual elements are the context providers. These contextual elements include the potential fire threats, current rate of fires, number of people to be rescued, humidity and wind conditions, nearby police, and ambulances—the CMP act as a bridge to promote context-awareness between the context providers and consumers. The component conCQeng present inside the CMP is responsible for enabling the QoC and CoC related features. The definitions, formulas, and functionalities of the active components in the framework are given below.

Relative Reputation: The satisfaction rate of a particular context consumer with the QoC delivered by a capable and available context provider.

RR Unit: Its value is measured as 0 or 1, representing the positive and negative QoC validity outcomes of the provided context concerning a particular context request from the consumer. Each RR unit contributes to the RR value assessment of the context provider in the selected RR collection (explained below).

RR Value: It represents the degree of Relative Reputation of a context provider in a particular RR collection, measured as an average using the formula given below.

$$\boldsymbol{RRCP(RRvalue)_{x+1} = \frac{RRCP(RRunit_{sum})_x + RRunit_{x+1}}{n+1}} \qquad [1]$$

Where,
RRCP(RRvalue)$_{x+1}$ is the updated RR value of the selected context provider,
RRCP(RRunit$_{sum}$)$_x$ is the sum of RR units of selected context provider in the RR collection,
RRunit$_{x+1}$ is the RR unit value generated for the new request,
n is the count of RR units in the RR collection.

RR Collections: The individual data records containing the elements to assist in QoC and CoC-aware selection of context providers to serve a particular request from the context consumer. These collections are formed and updated based on context providers' service eligibility and QoC validity. Each RR collection contains the context provider's SLA and RR details. The eligibility is determined based on SLA details: entity, contextual element, QoC elements and their values, cost, and QoC violation penalties. In contrast, the RR details: context consumer's Id, number of RR units generated, Sum of RR units, RR value, are the elements related to the QoC satisfaction rate of the context provider. Their values are updated based on the QoC provision to the query associated with RR collection.

Assurance: A method of sorting context providers with equivalent RR in the order of the low cost and high QoC violation penalties.

RR Processor and Repository: These components are responsible for performing assessment and selection of RR collections. The assessment includes two processes. The first is RR initiation; it is performed each time an eligible context provider is discovered for a particular customer request. The RR processor initiates the collection in RR repository by storing SLA and RR details as discussed in *RR collections*, with the initial *RR value* as 1, measured on the scale of 0-1. The second one is RR assessment; it is performed each time the context provider serves the request represented by RR collection. Based on the received context, first, the QoC evaluator validates the QoC elements and generates an RR unit. Then, the RR processor assesses and updates the RR value as given in Equation 1.

In the RR based selection, the processor identifies the relevant RR collection(s) for the requests in the incoming context query, and context provider(s) possessing the highest RR value from these collections are selected. Firstly, the relevant collections are identified based on their eligibility. This is performed by matching the incoming requests with the requests represented by the RR collections. Later from the identified collections, the context providers(s) with the highest RR are selected for the assurance phase. Therefore, contributing to personalised QoC aware selection.

Assurance Processor and Cache: The Assurance processor receives the equivalent context providers (providers with matching RR) from the RR processor with their cost and QoC violation penalty details. Using these details sorts them in the Assurance cache in the order of low costs and high penalties, and the first provider is invoked. Hence, contributing to profitability by providing the CoC awareness. It also receives the QoC violation alerts from the QoC validator and immediately invokes the succeeding context provider. Thus, preserving the platform's credibility by the failure safety. The Assurance cache is cleared only after the successful service completion.

QoC Evaluator*:* It is responsible for performing the QoC elements' measurement and validation, generating and sending RR unit values to the RR processor, and alerting the Assurance processor in case of QoC violations. In QoC measurement, the evaluator uses the context elements and their QoC meta-data (represents the provided context properties, for example, generation timestamp) to compute their QoC elements' values. The QoC validation is performed in two phases. The first phase occurs before context delivery; here, the evaluator validates the QoC by matching the generated QoC values with their requirements in the query. The second phase occurs after the delivery; here, if the consumer is not satisfied by the context, they report it back to the QoC evaluator. Based on the validity of the QoC elements from these two phases, the RR unit is generated and sent to the RR processor. Finally, in the event of invalid QoC, the QoC evaluator notifies the Assurance processor to invoke the next provider from their sorted list.

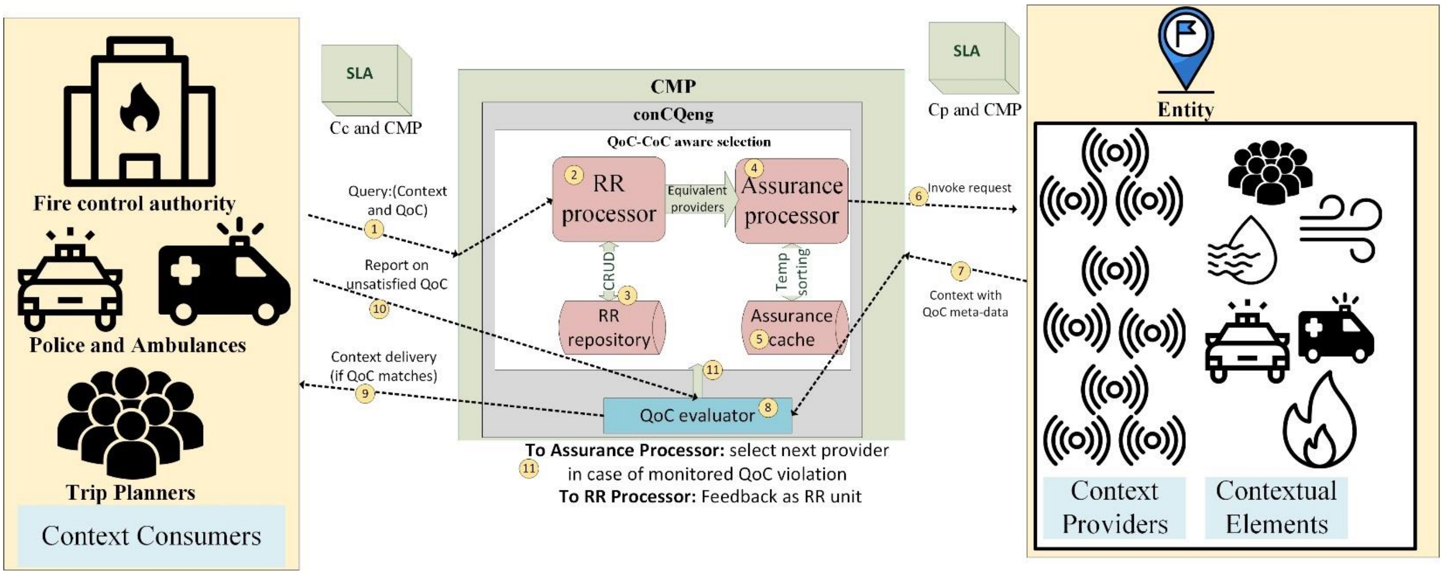

Figure 1. *The conceptual architecture of conCQeng.*

The system's process flow occurs according to the numbers given in Figure 1. In step 1, The context consumer initiates the process by sending a push-based context query that contains the context request(s) to the CMP; for these queries, context acquisition and processing are performed when a related event is triggered (e.g., when the fires are detected). Each query may contain requests with the context element and their QoC elements' values. For example, a fire control room query may request all the context elements from the location on a fire alert with 80% accuracy. After receiving the query, the CMP completes all the initial processes (out of this paper's scope) and forwards the requests to the RR processor in conCQeng. Using this input, in steps 2 and 3, the RR processor finds the eligible RR collections from the RR repository, selects the provider(s) with the highest RR value, and sends them to the Assurance processor. Then, in steps 4 and 5, the Assurance processor sorts the Equivalent context providers to possess the lowest cost and highest penalties, stores them in the Assurance cache and invoke the first provider.

After invoking the context provider in step 6, their context output is provided to the QoC evaluator in step 7. Then, in step 8, QoC elements: freshness, representation format, and accuracy are measured and validated. If the context is valid, as given in step 9, it is provided to the context consumer. Then, if the consumer is unsatisfied with the service, as shown in step 10, they report it back to the QoC evaluator. Based on the total validity, in step 11, the QoC evaluator generates and sends the RR unit value to the RR processor; this updates the context provider's reputation accordingly. Also, the Assurance processor is notified by the QoC evaluator to select the next best context provider for an invalid context.

4. Conclusion and future work

In this paper, we have proposed the conceptual architecture of conCQeng that uses two interrelated novel approaches called Relative Reputation and Assurance to promote efficient and effective QoC and CoC aware selection in CMPs. Integrating this component in CMPs can increase their context consumer experience by performing a personalised selection, and valid QoC aligned context delivery. Besides, it also increases platforms' profitability by the inclusion of CoC-awareness in the selection. Our future works will focus on designing the theoretical aspects, implementing the system prototype, and evaluating our Relative Reputation and Assurance assertions. Further objectives include developing and integrating the conCQeng with CoaaS [3] to increase its functionality and uptake by developers of context-aware applications.

Bibliography

[1] BUCHHOLZ T., KÜPPER A., SCHIFFERS M., « Quality of Context: what it is and why we need it », *proc. 10th work. openview univ. assoc. ovua*, pp.1-14, 2003.

[2] BAZIRE M., BRÉZILLON P., « Understanding context before using it », *International and Interdisciplinary Conference on Modeling and Using Context*, pp. 29-40, Springer, Berlin, Heidelberg, 2005.

[3] HASSANI A., MEDVEDEV A., HAGHIGHI P.D., LING S., INDRAWAN-SANTIAGO M., ZASLAVSKY A., PRAKASH JAYARAMAN P., « Context-as-a-Service Platform: exchange and share context in an IoT ecosystem », *IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pp. 385-390, IEEE, 2018.

[4] MANZOOR A., TRUONG H-L., DUSTDAR S., « Quality of context: models and applications for context-aware systems in pervasive environments », *The Knowledge Engineering Review*, 29(2), pp.154-170, 2014.

[5] KOUAME K-M., MCHEICK H., AJAMI H., « Adaptive Mechanism Model for the Prevention of SLA Violation in the Context of COPD Patient Monitoring », *Symmetry*, 12(9), p.1575, 2020.

[6] NITTI M., GIRAU R., ATZORI L., PILLONI V., « The importance of the feedback », *20th Conference on Innovations in Clouds, Internet and Networks (ICIN)*, pp. 325-327, IEEE, 2017.

[7] JAVAID S., AFZAL H., ARIF F., ILTAF N., ABBAS H., IQBAL W., « Catswots: Context aware trustworthy social web of things system », *Sensors*, 19(14), p.3076, 2019.