

A Quality-Aware Context Information Selection Based Fuzzy Logic in IoT Environment

Farida Retima¹, Saber Benharzallah², Laid Kahloul¹, and Okba Kazar¹

¹Smart Computer Sciences Laboratory, Biskra University, Algeria

²Smart Computer Sciences Laboratory, Batna2 University, Algeria

Abstract: *In the last decade, several works proposed their own approaches about the context management for the Internet of Things (IoT). An important issue in such systems is faced by context data distribution with a sufficient level of quality i.e., Quality of Context (QoC). In this paper, a fuzzy logic-based framework is proposed which handles QoC evaluating within distributed context manager and context-aware applications. In addition, IoT contains massive context sources and data. From this issue, we use MapReduce skyline to speed up the computation and introduce parallelism in the processing. This article presents also a solutions provided by the Model Driven Approach (MDA) for modelling the captured context information from different context source.*

Keywords: *IoT, context manager, MDA, fuzzy logic, QoC, context source, map reduce skyline, application.*

Received February 14, 2018, accepted April 17, 2018

1. Introduction

The IoT is the concept that aims to extend the regular Internet to the real-world physical objects. Consequently, a large number of things can be, at any time during their life cycle, either temporarily or permanently, connected to the global network infrastructure [7].

In Internet of Things (IoT), there are numerous applications that are achieved by things collaboration. These applications use context information. There is a need to offer context information to the application layer in order to facilitate the development of new applications. This can be ensured by middleware. Such IoT middleware called context manager assumes to deal with a huge amount of heterogeneous data generated by devices, and transformed them to a high level of abstraction to provide users with valuable output. With the exponential increase in the number of context sources (ex: sensors, social network, web services. etc.,) which offer the same type of context information that satisfies the requirements of the applications.

It has become important to use Quality of Context (QoC) as an essential criterion to select the context source which ensures the quality required by application. In [32], we have a previous study entitled "A Comparative Analysis of Context-Management Approaches for the Internet of Things" which published by The International Arab Journal of Information Technology (IAJIT).

In this study, we evaluated the most popular context management approaches in the IoT using on a set of criteria: heterogeneity, mobility, the influence of the physical world, scalability, security, privacy, QoC,

autonomous deployment of entities, characterization multi scales, interoperability, context acquisition, context modeling, context reasoning, context distribution, design method, and tools of implementation proposed for this purpose [32].

An important issue like the criteria of the QoC is missing in the most of the studied approaches. They don't handle any solution for the QoC that leading an application for good decisions and relevant reactions. There are a few works have presented reviews for managing the QoC [1, 3, 11, 13, 22, 28, 30, 31, 42].

In [22], the authors proposed a generic, expressive and computable QoC Information Model (QoCIM) to handle any QoC criterion within distributed context managers and QoC-aware applications in IoT. QoCIM eases the implementation of generic QoC transformation functions. The purpose of these functions is just to transform low-level information into high-level abstraction information. However, these functions do not reflect the fact of uncertainty of QoC provided by different context sources.

Also, this work does not handle an effective solution for increasing number of data to improve the computation of these functions. In [3, 31, 42], the authors deal only with categorizations and quality modeling in context information to facilitate the programming of context aware applications in pervasive environment. These works do not express any method for evaluating with various criteria of QoC.

In [30], the authors identify mechanisms such as probabilistic logic, fuzzy logic and Bayesian networks for dealing only with uncertainty criteria. However, this method is not enough to really express the overall

quality of information, whereas there are a various criteria which fully represent the QoC.

In [13, 28], the authors used the ontology to model context and its quality in a formal way. This solution achieves a shared semantic understanding of concepts and the relationships that hold among them. Such method is centralized and requires significant computing and storage resources. In [1], the authors are based on the integration of QoC management within the COSMOS platform for ambient environment.

It is a process-oriented based on a set of software components (Context Node, QoC Node) which are organized in hierarchical architecture. However, this method requires time and resources to create a new component of context information and its quality or to update one that already exist. This work does not allow efficient management of context quality because it deals with QoC as a component which has “input-output” and does not handle the imprecision of QoC description provided by different context sources. In [11], the authors presented a technique to combine different QoC metrics to infer the value of confidence on context using fuzzy logic. This work does not handle a solution for increasing number of data and the QoC to improve the computation of their confidence inference system.

Based on our comparative study, we consider the problem of selecting a quality of context according to consumer’s satisfaction-based preferences. However, the fast increasing number of context sources and data, multiple QoC attributes to be considered, pose a big challenge in IoT. To address this issue, in this paper, we propose a MapReduce skyline framework to speed up the computation and introduce parallelism for processing large amounts of context sources (described in Algorithm 2). Moreover, another feature of our proposed approach is using a fuzzy logic-based framework within a context manager (described in Algorithm 3). This feature aims to evaluating the QoC provided by each context source and QoC required by application. Our proposed solution is presented for essential reason: we apply a fuzzy logic to a small number of context sources resultant from map reduce skyline rather than a large number. Therefore, it selects optimal context sources from a large number of context sources by reducing the computation time and cost.

The use of fuzzy-logic is motivated by its problem-solving methodology that can be implemented on any management system regardless of size, complexity of the problem [20]. This solution is an appropriate for being used in representation of non-functional attribute description. It looks like human reasoning for using approximate information provided by different context sources. It can effectively deal with the inherent vagueness and imprecision of QoC description both on the client and on the provider side: The fuzzy sets are suited to specify both the QoC requirements raised by

an application and an approximate estimate which a provider can provide. Fuzzy logic was used to help the consumer in selecting the most matched quality of context through giving weights to available context sources.

The contribution of this paper addresses four issues: First, we focus on non-functional parameters of QoC in information description. Second, we propose a generic and extensible way to model the context information and the level of its quality which provided by different context source following a Model Driven Approach (MDA). Thirdly, we present a system model for implementing the skyline operator using MapReduce to handle massive data. Fourthly, we define a fuzzy-based framework for delivering the suitable QoC to user. Finally, we apply our approach in an illustrative example and we demonstrate its validity.

The rest of this paper is organized as follows; In section 2, we formally describe our approach and we prove our choices by a comparative study between different methods exist in literature in section 2.2. The description of our solution is discussed with an illustrative example in section 3. Lastly, we conclude the paper in section 4.

2. Proposed Approach

Our approach shows that a fuzzy-based framework, Map reduce skyline and the methodology of MDA are feasible and necessary for supporting quality of context modeling in IoT.

2.1. Model-Driven Approach

MDA is more suitable for the development of context-aware applications and context management system. Such an approach could facilitate developer’s task at design time, by providing specialized modeling languages, Metamodels and code generation tools [34]. MDA provides a high level of abstraction and it makes available tools and grammars.

This approach allows the construction of context models which may be used to model context information and can be automatically transformed to particular target platforms [39]. The modeling language used in the majority of cases is the Unified Modeling Language (UML) [4].

2.2. Map Reduce Skyline

MapReduce is a programming model and an associated implementation for processing and generating large data sets [10]. Programmers are required to execute only two functions, namely, the map (mapper) and reduce functions (reducer). The skyline is a set of points $SKY(S) \subseteq (S)$ which are not dominated by any other points. The points in $SKY(S)$ will be called skyline points [37].

We define our dominance relationship (context

source dominance) as follows: Context source CS_i dominates context source CS_j if and only if CS_i is not worse than CS_j for all QoC criteria and CS_i is better than CS_j for at least one QoC criteria.

To process efficiently skyline computation over a set of N servers in parallel, it is important to partition the dataset to the N servers.

From this issue, we do a comparison study which shows the differences between our proposed partitioning method and other methods exist in literature; our partitioning method based on partitioning the context sources to essential category according to their type (sensor, software component, service component, social network). This method is easy to implement and does not add any computational overhead in order to decide in which partition each context source falls. Generally, the performance cost in this method is better than other partitioning methods. In the work [15, 37] cited angle based partitioning method which Maps Cartesian coordinate space into hyper-spherical space.

The data space is partitioned into N partitions on the angular coordinates. The local skyline points reported in this approach are more likely to be global skyline points. This feature leads to smaller network communication costs and smaller processing costs for the merging phase.

But, The processing cost to determine the partition boundaries is not minimized and the fairness (the workload is evenly distributed among all available servers) of this technique depends on the boundaries of each partition, which in turn influence the overall performance of the parallel skyline computation.

In [37], described grid based partitioning method which is based on recursively dividing dimensions of the data space and it defines the boundaries of the partitions in each dimension. Moreover, each partition has approximately the same cardinality of data points; therefore the workload of each server is balanced. But, the server corresponding to the origin of the axes contributes the most to the result set, while several others, especially those far away from the axes, do not contribute at all. Also, each server returns, roughly, an equal amount of local skylines, but most of them do not contribute to the global skyline result set. The merging phase in this method has redundant workload and the communication cost is not minimized.

In [40], cited random partitioning method which is partitioned the data points randomly among the servers (each partition holds a random sample of the data). This method is easy to implement and the performance of random partitioning degrades significantly in the case of specific data distributions. However, the size of the local result sets is not minimized and maybe there are points that belong to the local skyline sets do not belong in the global skyline result set.

A number of algorithms have been proposed in the literature for answering skyline computation; we

choose Block Nested Loops (BNL) [37] to do local skyline computation in our solution (described in Algorithm 1). Each data object in this algorithm is compared with every other object in the database. The object is reported only if no other object dominates it.

BNL is widely accepted and easily to be realized compared with other algorithms which are relatively complex. It is strictly correct and complete and it works for all types of domains.

We do a comparison study which shows the differences between BNL and other methods exist in literature: Divide and Conquer (D and C) method [37] divides recursively the entire dataset into several blocks. D and C has a reasonable performance but it does not scale well for large datasets. Sort First Skyline (SFS) method [9] based to presorting the input dataset in an ascending order according to a monotone preference function. This method returns skylines points in a progressive way and decrease the number of comparisons between points. But, it cannot adapt to different user preferences and has to scan the entire dataset to return a complete skyline.

Linear elimination sort for skyline [12] algorithm is an optimized version of SFS which achieves a better average performance. But, it suffers the large number of pre-calculations required during the filter phase Skylines. Bitmap [15] algorithm encodes all data into a bitmap structure. The Skyline points are then obtained using only binary operations on the bit vectors. Bitmap is a progressive algorithm which means that it does not need to scan the complete dataset in order to return results. But, each addition, deletion or modification of a dimension or of a point involves the recalculation of all bit vectors. Nearest Neighbor (NN) [15] method is based on nearest neighbor using a given distance (e.g., Manhattan distance). It uses dynamic index structures (R-Tree), which can be incrementally updated.

However, NN has large input and output overhead, especially in high dimensional spaces, due to the recurrent access of the R*-tree. Branch and Bound Skyline (BBS) [27] apply nearest neighbor method to progressively output skyline points from datasets that are indexed by R*-trees. It guarantees the minimum inputs outputs cost and equivalently R-tree page accesses. But, the performance of BBS mainly depends on the size of computed skyline and the dimensionality of data.

2.3. Functioning of our Context Manager

Our approach is not focused on providing a complete software solution that addresses all the requirements needed in IoT paradigm. Instead, we focus on a single problem that is managing the QoC information.

We present our approach: distributed context architecture with components for collection, processing of context information. Our context manager focuses on context processing assuming all the context data are

produced by different context source. As a result, context architecture depicted in Figure 1 intended only to address that specific problem.

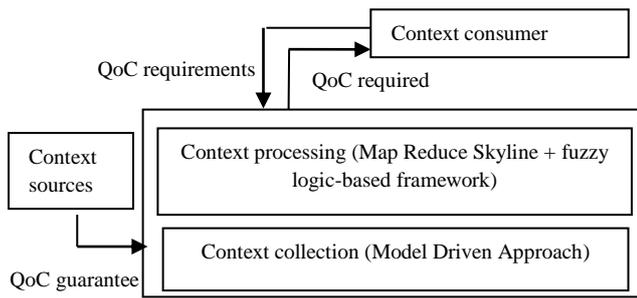


Figure 1. Architecture system.

Our architecture consists of two layers:

1. Context collection layer.
2. Context processing layer.

Most of the context management middleware always have similar layers with similar principal. We propose a quick walkthrough of these layers:

1. *Context Collection*: This layer appears in most the context management middleware solutions for IoT with different terminologies. This middleware are responsible for collecting the low-level context information from various available sources [21]. These context information and context sources characteristics need to be modeled in this layer. A model at this level will apply to a specific context source to provide a generic modeling solution for expressing the context provider. Our solution follows the MDA to model context provider.

The MDA approach aims to separate the design into two levels:

1. At the top level Platform Independent Model (PIM) which describes the software architecture of the system, but without taking into account the specificities of a particular platform. In the Figure 2 we describe our Meta model to illustrate our PIM which can be used to model the context source.
2. In a lower level, Platform Specific Model (PSM) is an extension of the PIM with concepts specific to a target platform. In this layer, the Model To Text (M2T) transformation extracts the source code from a template [33].

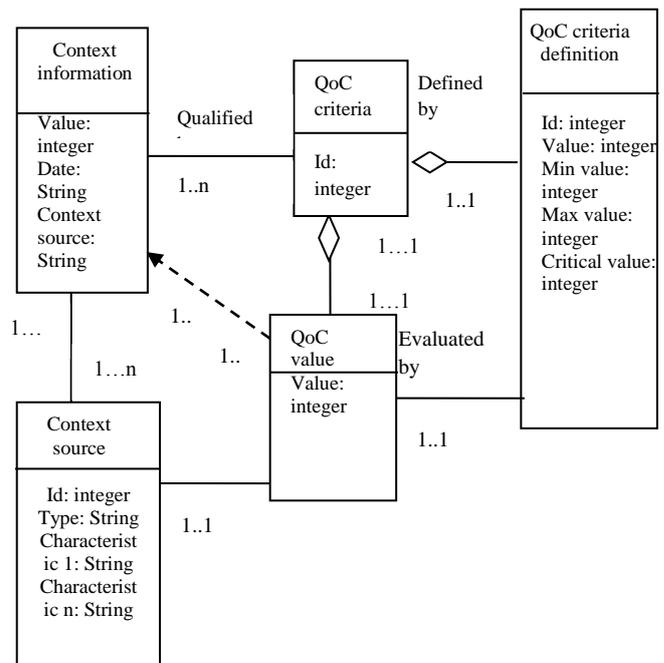


Figure 2. Context source meta-model.

2. *Context Processing*: The context processing layer is responsible for extraction of context information from context collection layer [21]. We develop in this layer a parallel program to process large amounts of datasets (context sources) to improve the efficiency by employing the MapReduce skyline paradigm. Then, we applied the fuzzy logic on final result list generated by MapReduce skyline.
3. *Application Side*: Initially, the end user gives request to context manager to find the list of context sources that are available to satisfy its need. User requirements are generally non-quantifiable. These requirements can be captured based on linguistic variables.

QoC for application request is defined by three elements:

- a. *Quality Criteria*: is the quality criteria in application requests.
- b. *Weigh Criteria*: design the importance which requester assigns to the quality criteria.
- c. *Quality Expectations*: is requester's expectations of the quality parameter based on linguistic terms.
4. *Context Source Side*: QoC for context source guarantee is defined by two elements: quality criterion in context source guarantee, and the level of quality of the context data that the producer is able to provide. This level of quality takes a value between [0, 1].

We first assess context sources qualities (precision, completeness, etc..) according to the properties of each context sources. Each QoC criteria takes a value between [0, 1]. Then, we propose to apply two methods (described in Algorithms 2 and 3):

- First Method: MapReduce skyline consists of two processes:

1. *The Map Process*: context sources are partitioned by the master server into multiple data blocks based on the various QoC criteria. There are many partitioning techniques in order to divide the data to be processed in parallel (our partitioning method compared to other methods which described in section 2.2). Then, the local skylines from context sources are generated in subdivided data blocks.
2. *The Reduce Process*: this function computes the global skylines from the local skylines generated in map process.

- Second Method: fuzzy logic:

We apply fuzzy logic on global skylines list generated in the first method. The fuzzy logic transforms the QoC criteria values of each context source in the global skylines list into fuzzy values using membership functions. The result of these membership functions allows the inference rules to derive the utility values of each context source.

We describe these two methods by applying our proposed algorithms (Algorithms 2 and 3). For this purpose, we adopt on the BNL algorithm (Algorithm1) proposed by Borzsonyi *et al.* in [5] as follows:

Algorithm 1: SkylineBNL

Input: input of the Skyline operation (set of points M)
Output: output of the Skyline operation (set of points R)
Begin algorithm 1
 // T: temporary file, S: main memory, $p < q$: point p is dominated by point q;
 // Initialization: $R: = \emptyset$, $T: = \emptyset$, $S: = \emptyset$; $CountIn: = 0$; $CountOut: = 0$;
 While: $\neg EOF(M)$ do begin
 For each $p \in S$ do // transfer points which have been compared to all point;
 If $TimeStamp(p) = CountIn$ then save(R,p), release(p);
 Load (M,p), $TimeStamp(p) = CountOut$; //load next point
 $CountIn: = CountIn + 1$;
 For each $q \in S \setminus \{p\}$ do begin // compare p with all points q;
 If $p > q$ then release (p), break;
 If $p < q$ then release (q);
 End for
 If $\neg MemoryAvailable$ then begin
 Save (T,p), release(p);
 $CountOut: = CountOut + 1$;
 End if
 If $EOF(M)$ then begin
 $M: = T$, $T: = \emptyset$; $CountIn: = 0$, $CountOut: = 0$
 End if
 End while
 For each $p \in S$ do save (R,p), release(p);
 Return R;
 End Algorithm 1

Our proposed algorithms as follows:

Algorithm 2: Map reduce skyline

Input: Different Context Sources (CS) (sensors, social network, software applications, web services); Output: The list of global skylines (GS);

Begin Algorithm 2

// Apply our proposed partitioning method which based to classify the context source into categories (Sensor, Software component, service component, Social Network) to obtain the partitions list P_i (Sensor partition, Social Network partition, Software component partition, Service component partition);

For each $CS_i \in CS$ do Begin

If ($CS_i = \text{Sensor}$) then Sensor partition $\leftarrow CS_i$;

Else begin

If ($CS_i = \text{social network}$) then Social Network partition $\leftarrow CS_i$;

Else begin

If ($CS_i = \text{software}$) then Software component partition $\leftarrow CS_i$;

Else

If ($CS_i = \text{service}$) then Service component partition $\leftarrow CS_i$;

End else

End else

End for.

For each partitioned block P_i (context source category) do begin

// Compute Local Skyline LS_i from context sources CS using BNL described in algorithm 1);

$LS_i \leftarrow BNL(P_i)$;

Output (P_i, LS_i) End for

// Compute the Global Skylines GS from the Local Skylines LS_i using BNL);

$GS \leftarrow BNL(LS_i, \dots, LS_n)$;

Output (GS);

End Algorithm 2

Algorithm 3: Fuzzy logic

Input: GS;

Output: Weigh of each context source in global skyline list based on the linguistic terms (WCS);

Begin

// Apply the fuzzification using one of the membership functions (ex: triangular membership functions);

For each $CS_i \in GC$ (context source in global skyline) do

// Fuzzification (input: QoC Criteria values (QC), output: QoC Criteria based on the linguistic terms (QC_linguistic));

$QC_linguistic \leftarrow Fuzzification(QC)$;

// Inference rules;

$WCS \leftarrow Inference\ rules(QC_linguistic)$;

End for.

End

Algorithm 3// context source which has the higher weigh is the selected candidate;

The first proposed algorithm is based on partitioning the grand number of context sources which provide the same type of context information requested by an application. Our partitioning method is based to classify the context source into categories. Then, we compute the local skylines of each context source category (context sources are not dominated by any other context source in partition P_i) according to application preferences by using the method skyline BNL.

Finally, local skylines are merged into a global skyline by using also method skyline BNL.

In the second proposed algorithm, we apply the fuzzy logic only on the global skylines list which has

the most matched with application request from context sources. Due to our generated inferences rules, we select the context source which has the higher weigh to satisfy the need of application.

3. An Illustrative Example

QoC criteria indicate the quality of context information from different aspects by different parameters. There are many researches in literature that defined QoC metrics are precision [25], Accuracy [18], Timeliness [18], Up-to dateness [6], Coverage [25], Trust-worthiness [6], Completeness [16], Probability of Correctness [16], Reliability [18], Significance [11], Usability [18], Representation Consistency [11], Access Right [18] and Granularity [18].

At first, we introduce some research for computing the QoC metrics according to context source type as follows in Table 1:

Table 1. Evaluation methods for QoC metrics.

Context sources	Evaluation methods of QoC metrics
All kinds of Sensors [11]	Dustdar and Manzoor [11] focus on evaluation of QoC metrics for sensors by using many equations according to the properties of sensors.
Any Social Network [36]	Soler computed QoC metrics for social network according to the properties of this network (The size of the network, The stability of social support...), and the Degree centrality(is a measure that indicates how well connected an actor is within the network).
Any service component [24]	Menasce [24] described the non functional characteristics of web services. The quality of information provided by web service corresponds to the quality of service QoS.
Any Software component [35]	Sharma <i>et al.</i> [35] uses Analytical Hierarchy Process (AHP) to assign the weight values to the characteristics for the proposed model. These weight values are then used to evaluate the quality contribution of sub-criteria, criteria and finally, the overall quality of the software component by using the appropriate metrics.

We choose these four methods for QoC measurement of each context source category because they have a low processing cost and they are easy to use compared to other methods. We consider following QoC criteria in this example just to clarify our proposed solution: Precision, Accuracy. Let us assume that Table 2 shows the calculated QoC criteria values by using the methods described in Table 1.

Table 2. QoC criteria values.

Context sources available	Precision	Accuracy
Sensor 1	0.5	0.2
Social Network 1	0.8	0.1
Sensor 2	0.6	0.4
Software 1	0.3	0.6
Social Network 2	0.4	0.3
Sensor 3	0.1	0.8
Software 2	0.9	0.5
Software 3	0.7	0.5

In order to verify our solution, we apply Algorithm 2 (MapReduce skyline) in our example data. The first

step of the map reduce skyline is using one of the partitioning methods. We propose in this example the partitioning according to context source category (sensors, software component, service component, social network).

In this method, each partition processes the query independently. In this case, we obtain that: Sensor partition P1= {sensor 1, sensor 2, and sensor 3}, Social Network partition P2= {social network 1, social network 2} and Software component partition P3= {software 1, software 2, and software 3}.Then, we compute within each small partition the local skylines context sources by using BNL. Consequently, we focus on how to perform distributed computation skyline to retrieve the meaningful subset of local skyline context sources of each partition according to user preference.

We need to address the order of criterions according to their importance. The importance of each criteria stems from the application request. The more important criteria are the higher priority which should take into account and the others are negligible.

In our example, Assuming that an application is interested about information which has a high precision and medium accuracy and these two QoC criteria has an equal importance. In this case, the local skylines based on the BNL for the Sensor partition P1, Social Network P2, and Software component P3 is interesting about:

1. According to the dominance definition, a context source dominates another context source because it has higher precision and accuracy. Thus, the skyline points are the best possible tradeoffs between precision and accuracy.
2. Retrieve a context source as local skyline if it is better than or equal to any context sources in all QoC criteria.
3. Eliminate context sources dominated by others in the partition.

• The Local Skyline List is:

For Sensor partition P1: sensor 2 and sensor 3

For Social Network partition P2: social network 1 and social network 2.

For Software component P3: software 1, software 2.

So, the local skyline list LS after the computation is :{ sensor 2, sensor 3, social network 1, social network 2, software 1, and software 2}.

Then, the local skyline list LS are merged and integrated into a global skyline by using BNL. Finally, the global skyline GS list is :{ software 1, sensor 3, and software 2}.Based on the result of global skylines, we select the context source which is the most matched to request application by applying Algorithm 3 (fuzzy logic).

The QoC criteria values are fuzzified using three triangular membership functions: $F_H(x)$, $F_M(x)$, and $F_L(x)$.These membership functions are presented in

Equations (1), (2), and (3), respectively. They assess high, medium, and low membership values of QoC criteria, respectively [41]:

$$F_H(x) = \begin{cases} \frac{x-c}{d-c} & \text{if } c \leq x \leq d \\ \frac{e-x}{e-d} & \text{if } d < x \leq e \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

$$F_M(x) = \begin{cases} \frac{x-b}{c-b} & \text{if } b \leq x \leq c \\ \frac{d-x}{d-c} & \text{if } c < x \leq d \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

$$F_L(x) = \begin{cases} \frac{x-a}{b-a} & \text{if } a \leq x \leq b \\ \frac{c-x}{c-b} & \text{if } b < x \leq c \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

Where the constants: *a*, *b*, *c*, *d* and *e* are provided by an expert depending on the data. This function is simple and it takes the user values, then map directly to the fuzzy set. For systems that need significant dynamic variation in a short period of time, a triangular should be utilized [41]. If we put: *a*=0; *b*=0.1; *c*=0.6; *d*=1; *e*=0.3, we obtain the following fuzzy values (described in Table 3):

Table 3. Calculated fuzzy values for global skylines.

Global skyline list	Precision	Accuracy
Software 1	0 % high 40% medium 60% low	0 % high 100 % medium 0 % low
Sensor 3	0 % high 0 % medium 100 % low	50 % high 50 % medium 0 % low
Software 2	75 % high 25 % medium 0 % low	0 % high 80 % medium 20 % low

The next step of fuzzy logic is the inference rules which are written by the designer of fuzzy system based on knowledge that he has. We perform inference rules for software 1, sensor 3 and software 2. The inference rules generated are:

- For software 1:

IF Precision 0% high AND Accuracy 0% high THEN (Weight-Software1) 0% high.
 IF Precision 0% high AND Accuracy 100% medium THEN (Weight-Software1) 0% high.
 IF Precision 0% high AND Accuracy 0% low THEN (Weight-Software1) 0% low.
 IF Precision 40% medium AND Accuracy 0% high THEN (Weight-Software1) 0% high.
 IF Precision 40% medium AND Accuracy 100% medium THEN (Weight-Software1) 40% medium.
 IF Precision 40% medium AND Accuracy 0% low THEN (Weight-Software1) 0% low.
 IF Precision 60% low AND Accuracy 0% high THEN (Weight-Software1) 0% high.

IF Precision 60% low AND Accuracy 100% medium THEN (Weight-Software1) 60% low.
 IF Precision 60% low AND Accuracy 0% low THEN (Weight-Software1) 0% low.

- For sensor 3:

IF Precision 0% high AND Accuracy 50% high THEN (Weight-Sensor 3) 0% high.
 IF Precision 0% high AND Accuracy 50% medium THEN (Weight-Sensor 3) 0% high.
 IF Precision 0% high AND Accuracy 0% low THEN (Weight-Sensor 3) 0% low.
 IF Precision 0% medium AND Accuracy 50% high THEN (Weight-Sensor 3) 0% medium.
 IF Precision 0% medium AND Accuracy 50% medium THEN (Weight-Sensor 3) 0% medium.
 IF Precision 0% medium AND Accuracy 0% low THEN (Weight-Sensor 3) 0% medium.
 IF Precision 100% low AND Accuracy 50% high THEN (Weight-Sensor 3) 50% high.
 IF Precision 100% low AND Accuracy 50% medium THEN (Weight-Sensor 3) 50% medium.
 IF Precision 100% low AND Accuracy 0% low THEN (Weight-Sensor 3) 0% low.

- For software 2:

IF Precision 75% high AND Accuracy 0% high THEN (Weight-Software2) 0% high.
 IF Precision 75% high AND Accuracy 80% medium THEN (Weight-Software2) 75 % high.
 IF Precision 75% high AND Accuracy 20% low THEN (Weight-Software2) 20% low.
 IF Precision 25% medium AND Accuracy 0% high THEN (Weight-Software2) 0% high.
 IF Precision 25% medium AND Accuracy 80% medium THEN (Weight-Software2) 25% medium.
 IF Precision 25% medium AND Accuracy 20% low THEN (Weight-Software2) 20% low.
 IF Precision 0% low AND Accuracy 0% high THEN (Weight-Software2) 0% low.
 IF Precision 0% low AND Accuracy 80% medium THEN (Weight-Software2) 0% low.
 IF Precision 0% low AND Accuracy 20% low THEN (Weight-Software2) 0% low.

We based in our inference rules to AND operator which corresponds to the MIN operator. In our inference rules, we have several rules that generate several values of the same linguistic variable, we can choose an operator OR to combine the values of the same variable.

For example, we have four rules that generate the linguistic variable “high”: weight-Software2 is high to 0% and 75%. If we use an operator OR (Maximum operator), the variable “weight-Software2 is high” will have a final value of 75%. Therefore:

- For software2, we obtain: weight-Software 2 is high to 75%, medium to 25% and 0% low.

- For sensor 3, we obtain: weight-Sensor 3 is high to 50%, medium to 50 % and 0% low.
- For software 1, we obtain: weight-Software 1 is high to 0%, medium to 40 % and 60% low.
- Finally, the software 2 has the higher weigh so it is the selected candidate.

4. Conclusions

In this paper, a fuzzy logic-based framework was proposed for QoC management in IoT. We use QoC parameters for context sources selection. The selection process allows getting the best-fitting information which satisfy user request. This process evaluates QoC criteria of context sources and context consumers. Moreover, in our proposed approach we consider the parallel distributed MapReduce skyline to speed up the computation process and to handle massive context sources and data. Furthermore, in this framework, we use a solution for designing context source and define their captured information following a MDA. This last is best suited for developers with independence between the business logic and the technological aspect. Our future work is to apply the proposed approach on a real dataset of context source guarantee and application requirements.

References

- [1] Abid Z., Gestion de la Qualité de Contexte Pour L'intelligence ambiante," Ph.D Thesis, Tel'ecom SudParis et Universit ' e d' ' Evry-Val-d'Essonne, 2012.
- [2] Arcangeli J., Camps V., Desprats T., Laborde R., Lavinal E., Peninou A., Zarate P., Boujbel R., Oglaza A., Marie P., Bouzeghoub A., Chabridon S., Conan D., Taconet C., Lim L., Marquez S., Mignard C., Rottenberg S., Leriche S., Mbarki M., and Maurel H., "Projet INCOME: INfrastructure de Gestion de COntexte Multi-Echelle pour l'Internet des Objets," in *Proceedings of Conférence Francophone sur les Architectures Logicielles*, Paris, pp. 1-2 , 2014.
- [3] Becker S., Blessing A., Durr F., and Rothermel K., "Reference Model for the Quality of Context Information," Technical Report, 2010.
- [4] Borges V., "Survey of Context Information Fusion for Ubiquitous Internet-of-Things (IoT) Systems," *Open Computer Science Journal*, vol. 6, no. 1, pp. 64-78, 2016.
- [5] Borzsony S., Kossmann D., and Stocker K., "The Skyline Operator," in *Proceedings of the 17th International Conference on Data Engineering*, Heidelberg, pp. 421-430, 2001.
- [6] Buchholz T., Kupper A., and Schiffers M., "Quality of Context: What It Is and Why We Need It," in *Proceedings of the 10th Workshop of the Open View University Association: OVUA'03*, Geneva, pp. 1-14, 2003.
- [7] Chabridon S., Laborde R., Desprats T., Oglaza A., Marie P., and Marquez S., "A Survey on Addressing Privacy Together With Quality of Context for Context Management in the Internet of Things," *Annals of Telecommunications Journal*, vol. 69, no. 1-2, pp. 47-62, 2013.
- [8] Chabridon S., Conan D., Abid Z., and Taconet C., "Building Ubiquitous QoC-Aware Applications through Model-Driven Software Engineering," *Science of Computer Programming Journal*, vol. 78, no. 10, pp. 1912-1929, 2013.
- [9] Chomicki J., Godfrey P., Gryz J., and Liang D., "Skyline with Presorting," in *Proceedings of International Conference on Data Engineering*, Bangalore, pp. 717-719, 2003.
- [10] Dean J. and Ghemawat S., "MapReduce: Simplified Data Processing on Large Clusters," *Communications of the ACM Journal*, vol. 51, no. 1, pp. 107-113, 2008.
- [11] Dustdar S. and Manzoor A., "Quality of Context in Pervasive Systems: Models, Techniques, and Applications," Ph.D Thesis, Technical University of Vienna, 2010.
- [12] Godfrey P., Shipley R., and Gryz J., "Maximal Vector Computation in Large Data Sets," in *Proceedings of the 31st International Conference On Very Large Data Bases*, Trondheim, pp. 229-240, 2005.
- [13] Gu T., Wang X., Pung H., and Zhang D., "An Ontology-based Context Model in Intelligent Environments," in *proceedings of Communication Networks and Distributed System Modeling and Simulation Conference*, Singapore, pp. 270-275, 2004.
- [14] Hoyos J., Molina J., Botia J., and Preuveneers D., "A Model-Driven Approach for Quality of Context in Pervasive Systems," *Computers and Electrical Engineering journal*, vol. 55, pp. 39-58, 2016.
- [15] Kalyvas C. and Tzouramanis T., "A Survey of Skyline Query Processing," *Computing Research Repository Journal*, 2017.
- [16] Kim Y. and Lee K., "A Quality Measurement Method of Context Information in Ubiquitous Environments," in *Proceedings of International Conference on Hybrid Information Technology*, Cheju Island, pp. 576-581, 2006.
- [17] Lim L., Marie P., Conan D., Chabridon S., Desprats T., and Manzoor A., "Enhancing Context Data Distribution for the Internet of Things Using Qoc-awareness and Attribute-Based Access Control," *Annals of Telecommunications journal*, vol. 71, no. 3-4, pp. 121-132, 2016.

- [18] Manzoor A., Truong H., and Dustdar S., "Quality of Context: Models and Applications for Context-aware Systems in Pervasive Environments," *The Knowledge Engineering Review journal*, vol. 29, no. 2, pp.154-170, 2014.
- [19] Marie P., *Gestion de bout en bout de la Qualité de Contexte Pour l'Internet des Objets: le Cadriciel QoCIM*, Doctoral Thesis, University of Toulouse, 2015.
- [20] Marie P., Desprats T., Chabridon S., and Sibilla M., "Extending Ambient Intelligence to the Internet of Things: New Challenges for QoC Management," in *Proceedings of the International Conference on Ubiquitous Computing and Ambient Intelligence*, Belfast, pp. 224-231, 2014.
- [21] Marie P., Desprats T., Chabridon S., Sibilla M., and Taconet C., "From Ambient Sensing to IoT-based Context Computing: An Open Framework for End to End QoC Management," *Sensors Journal*, vol. 15, no. 6, pp. 14180-14206, 2015.
- [22] Marie P., Desprats T., Chabridon S., and Sibilla M., "QoCIM: A Meta-model for Quality of Context," in *Proceedings of 8th International and Interdisciplinary Conference on Modeling and Using Context*, Annecy, pp. 302-315, 2013.
- [23] Marquez S., *Models and Algorithms for Managing Quality of Context and Respect for Privacy in the Internet of Things*, Thesis, University of Paris-Saclay, 2015.
- [24] Menasce D., "QoS Issues in Web Services," *IEEE Internet Computing Journal*, vol. 6, no. 6, pp. 72-75, 2002.
- [25] Nazario D., Andrade A., Borgers L., Ramos W., Todesco J., and Dantas M., "An Enhanced Quality of Context Evaluating Approach in the e-Health Sensor Platform," in *Proceedings of the 11th ACM Symposium on QoS and Security for Wireless and Mobile Networks*, Cancun, pp. 1-7, 2015.
- [26] Nazário D., Dantas M., and Todesco J., "Context Management: Toward Assessing Quality of Context Parameters in Ubiquitous Ambient Assisted Living Environment," *Journal of Information Systems and Technology Management*, vol. 11, no. 3, pp. 569-590, 2014.
- [27] Papadias D., Tao Y., Fu G., and Seeger B., "An Optimal and Progressive Algorithm for Skyline Queries," in *Proceedings of International Conference on Management of Data*, California, pp. 467-478, 2003.
- [28] Preuveneers D. and Berbers Y., "Quality Extensions and Uncertainty Handling for Context Ontologies," in *Proceedings of Context and Ontology Conference*, Trentino, 2006.
- [29] Priya N. and Chandramathi S., "QoS Based Optimal Selection of Web Services Using Fuzzy Logic," *Emerging Technologies in Web Intelligence Journal*, vol. 6, no.3, pp. 331-339, 2014.
- [30] Ranganathan A., Al-Muhtadi J., and Campbell R., "Reasoning about Uncertain Contexts in Pervasive Computing Environments," *IEEE Pervasive Computing*, vol. 3, no. 2, pp. 62-70, 2004.
- [31] Razzaque M., Dobson S., and Nixon P., "Categorization and Modelling of Quality in Context Information," *Trusted and Autonomic Computing journal*, 2006.
- [32] Retima F., Benharzallah S., Kahloul L., and Kazar O., "A Comparative Analysis of Context-Management Approaches for the Internet of Things," *The International Arab Journal of Information Technology*, vol. 14, no. 4A, pp. 578-585, 2017.
- [33] Rottenberg S., *Modèles, Méthodes et Outils Pour les Systèmes Répartis Multi échelles*, Thesis, Institut National des Télécommunications, 2015.
- [34] Rottenberg S., Leriche S., Taconet C., Lecocq C., and Desprats T., "MuSCa: A Multiscale Characterization Framework for Complex Distributed Systems," in *Proceedings of the Federated Conference on Computer Science and Information Systems*, Warsaw, pp.57-85, 2014.
- [35] Sharma A., Kumar R., Grover P., "Estimation of Quality for Software Components –an Empirical Approach," *ACM SIGSOFT Software Engineering Notes Journal*, vol. 33, no. 6, pp. 1-10, 2008.
- [36] Soler L., *Social Networks and Performance in Knowledge Creation. An Application and a Methodological Proposal*, Ph.D Thesis, University of Girona, 2006.
- [37] Stella M., *Parallel Skyline Computation on Map -Reduce Systems*, Ph.D Thesis, Technical University of Crete School of Electronic and Computer Engineering, 2013.
- [38] Suchithra M. and Ramakrishnan M., "Non Functional QoS Criterion Based Web Service Ranking," in *Proceedings of the International Conference on Soft Computing Systems*, India, pp. 79-90, 2016.
- [39] Taconet C. and Kasi-aoul Z., "Building Context-Awareness Models for Mobile Applications," *Digital Information Management Journal*, vol. 8, no. 2, pp. 78-87, 2010.
- [40] Vlachou A., Doukeridis C., Kotidis Y., "Angle-based Space Partitioning for Efficient Parallel Skyline Computation," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Vancouver, pp. 227-238, 2008.
- [41] Yahyaoui H., Almulla M., and Maamar Z., "A Fuzzy Model for Selecting Social Web Services," in *Proceedings of the International Conference*

on *Web Information Systems Engineering*, Wise, pp. 32-46, 2014.

- [42] Zimmermann A., *Context Management and Personalisation: a Tool Suite for Context-and User-Aware Computing*, the Association for Computing Machinery, 2007.



Farida Retima is a PhD student in intelligent Computer Science Laboratory University Mohamed Khider Biskra, Alegria.



Saber Benharzallah is an associate professor and researcher in the computer science department of Batna2 University, (Algeria). Received his Ph.D degree in 2010 from the same department. Dr Benharzallah is currently the team leader of semantic interoperability and their applications in information systems and web services, in LINFI laboratory (Biskra University).



Laid Kahloul received the Ph.D. degree in computer software and theory from the Computer Science Department, Biskra University, Biskra, Algeria, in 2012. He is currently an associate Professor with the Computer Science Department, Biskra University, Algeria.



Okba Kazar professor in the Computer Science Department of Biskra, he helped to create the laboratory LINFI at the University of Biskra. He is a member of international conference program committees and the "editorial board" for various magazines. His research interests are artificial intelligence, multi-agent systems, web applications and information systems.