International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 - 5161

Available at http://inpressco.com/category/ijcet

Research Article

Probabilistic Context Retrieval Time-Based Algorithms in Dynamic Environments

Ahmed. A. A. Gad-ElRab[Å], T. A. A. Alzohairy[Å] and Almohammady S. Alsharkawy[Å*]

[Å]Department of Mathematics,Faculty of Science, Al-Azhar University,Cairo, Egypt

Accepted 10 May 2014, Available online 01 June2014, **Vol.4, No.3 (June 2014)**

Abstract

In mobile environments, mobile devices move continuously and have limited resources as computation, energy, and memory. Data management is one of the biggest challenges in this environment. Using the context in mobile environments is receiving considerable attention to meet these challenges. In the context-aware systems, the application can uses the contextual information such as user's location, day time, nearby people and devices and user's activity in a useful way to solve some mobile environmental issues. Context retrieval problem is one of the key issues in context awareness computing. In this paper, we describe the terms of context-aware systems and context-aware middleware. Also, we give an exemplary overview about context's definitions and classifications. In addition to introducing new context classification approaches will introduce in this paper. Furthermore, this paper develops context retrieval approaches in mobile environments to reduce context retrieval energy consumption. Finally, the simulation results show that the proposed context retrieval approaches are much better than the traditional approach.

Keywords: *Context, Context-Aware Systems, Context Middleware, Context retrieval.*

1. Introduction

Context-aware computing is a rapidly growing field in ubiquitous computing. Which is concerning with the adaptation of mobile application to the changing of the surrounding environment and situations. Context-aware computing is a prosperous field of research involving communication engineering, computer science and information technology. More precisely mobile communication, Human Computer Interaction, wearable computing, augmented reality, data management, feature extraction, artificial intelligence and decision making.

Context-aware system uses the context attributes such as location, time, and user activity to obtain the best services to mobile user. The key objective of these systems is to significantly simplify computing devices usage by realizing the permanent changes in entity's status and the surrounding environments. In the current time there are many systems use context-aware by way of example, not exhaustive enumeration (Health care, Services discovering applications, Advertising, E-commerce, E-learning/M-learning, Tourism and Traveling, Office and other Business applications, Entertainment, Emergency applications, Smart environments, Gaming and Social community applications). Context-aware systems use the contextual information to clarify the current situation and adopt mobile system to be suitable for user and device requirements. Context attribute represent any single information about user, device or the surrounding environments.

Mobile environments represent what is in place and surrounding, including users, devices and other resources. These environments provide a challenging and realistic environment for the current systems development because of mobility, limitation of energy and computations and other resources of environment's entities. So, we cannot apply any of the standard approaches to develop a mobile system suited to these types of environments. In order to build mobile system compatible with these changeable environments, the developers must give due consideration to the importance of environmental context. Which allow these systems to be appropriate to the nature of the mobile environments which are located. Also, using the contextual information leads to increase the utility, the efficiency, and reliability of the mobile applications.

Data management is one of the biggest challenges in mobile environments. Utilizing the context in mobile devices is receiving considerable attention to meet these challenges. In context-aware systems, the mobile applications can use the contextual information such as user's location, day time, nearby people and devices, and user's activity in a useful way to solve many mobile system's issues. Context retrieval problem is one of the key issues in context-aware systems. This problem deals with retrieving the required context with minimal cost to management the data in these environments.

In this paper, we will discuss the context-aware systems, and surveys the previous context definitions and classifications approaches. Finally, we will propose new two context retrieval approaches to reduce context's retrieval cost.

The rest of the paper is organized as the following. Section 2 includes a detailed survey of the related work. Section 3 introduces new context classifications in mobile environments. Section 4 describes and formulates the

*Corresponding author: **Almohammady S. Alsharkawy**

context retrieval problem. Section 5 introduces the proposed context retrieval algorithms. Simulation results and analysis presented in Section 6. Finally, Section 7 concludes the paper.

2. Related Work

2.1 Context-Aware Systems

Mobile context-aware system is a system takes into consideration the user's current situation, device and the surrounding environments. Also, this system adapts its services according to the changes of contextual values. (Gregory D Abowd et al, 1999) provided a definition of context-aware system as:

"A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task".

The context-aware system must consist of two main separated phases, the first phase contains context-aware middleware and the second phase includes decision making and computational intelligent approaches and context-aware applications.

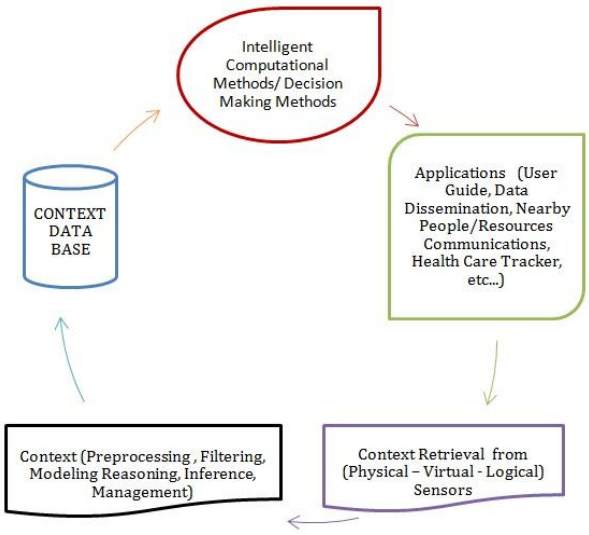

Fig.1 Context-Aware System Life Cycle

The context-aware middleware is responsible for retrieving context from sensors or any other data source, Context Preprocessing, Filtering, Context Modeling, Context Reasoning and Inference and Context Management. Context-aware middleware provides each context-aware application with the needed contextual information e.g. user's location, user's activity and environmental information. Fig-1 illuminates the phases of the context-aware system. (Peizhao Hu et al, 2008) introduced three main approaches that any developer can follow to build a context-aware application:

- **No application-level context model**: Applications perform all the actions, such as context acquisition, preprocessing, storing, and reasoning within the application boundaries.
- **Implicit context model**: Applications use libraries, frameworks, and toolkits to perform context acquisition, preprocessing, storing, and reasoning tasks.
- **Explicit context model**: Applications use a context management infrastructure or middleware solution. Therefore, actions such as context acquisition, preprocessing, storing, and reasoning lie outside the application boundaries.

2.2 Context-Aware Middleware

In most cases the context-aware system acquires the context from mobile sensors or obtained directly from system's user. There are three types of sensors (Physical Sensors, Virtual Sensors and Logical Sensors). Sensed values may need to normalized or scaled or transformed into different value ranges or domains to be suitable to be used by the system. For example, an electrical signal from a temperature sensor gets mapped to a temperature value on a Celsius or Fahrenheit temperature scale.

(Peter Brown et al, 2000) have classified context-aware applications into six types: proactive triggering, streamlining interaction, memory for past events, reminders for future contexts, optimizing patterns of behavior, and sharing experiences. The authors in, (Matthias Baldauf et al, 2007) described the context-aware middleware layers as follows.

- **Context retrieval layer**: Responsible for retrieving a raw of context data. Retrieving context means collecting context from different sensors. Sensors not only a hardware but also to any data source which may provide usable context such as virtual or logical sensors.
- **Context preprocessing layer**: Responsible for filtering, reasoning and interpreting context. The sensors queried in the underlying layer mostly often return technical data that are not appropriate to be used by application. Preprocessing layer also, is responsible for the extraction and quantization operations. For example, converting GPS position to the place name where the person is.
- **Context management layer**: Organizes the gathered data and offers them via a public interface to the client. (Stefan Poslad, 2009) Indicates that context management should include functions of discovery, storage, support, and access control. Fig-2 illustrates context middleware layers.

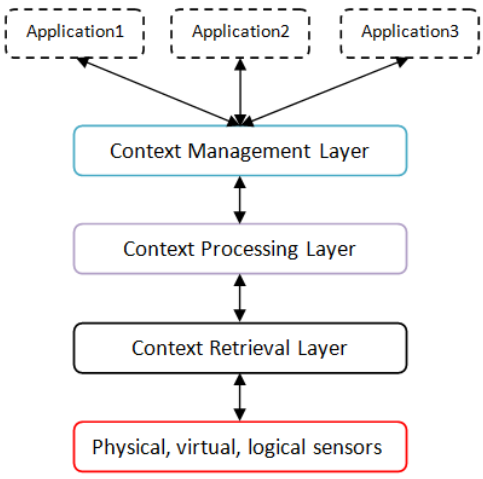

Fig. 2: Context-Aware Middleware Layers

3. New Context Classifications in Dynamic Environments

In this section, we will define the term of context and introduce new context classification approaches that will help to improve the context-aware systems development in the mobile environments.

3.1. Context Definition

Defining context is a good start to build an efficient system in mobile environments. So, if we have a well defined context term, we will be aware of all constrains that we face in developing any context-aware system. Also, defining the context will help the system's developer to know if the context will be suitable to build a reliable system in these environments or not. (Gregory D Abowd et al, 1999) introduced a context definition as follows:

"Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves".

The context attribute represents any single information about user, device or the surrounding environment. (Charith Perera et al, 2013) defined a context attribute as follows:

"A context attribute is an element of the context model describing the context. A context attribute has an identifier, a type and a value, and optionally a collection of properties describing specific characteristics"

According to (Gregory D Abowd et al, 1999), the previous context definitions were based on two different perspectives.

1) **Definition by synonymous**: Is to define the context by a sentence that explains the concept of context word, the information the context will represent, the entities or things covered by context. These types of context's definitions will be found in (Fausto Giunchiglia; Bill Schilit; Andy Ward; David Franklin; Jason Pascoe; Tom Rodden; Gregory D Abowd; Anind K Dey; Henry Lieberman; Guanling Chen; Karen Henricksen; Gregory D Abowd,2000; Paul Dourish; Ling Feng; Sungjin Ahn; Andreas Zimmermann; C. Bolchini; Constantin Schmidt; Michael Knappmeyer; CharithPerera).
2) **Definition by example**: Is to express the word of context via examples of fact words that have the same meaning of context or the attributes that constitute the context. These types of context definitions will be found in (Bill N Schilit; Randy H Katz; CE Billings; Richard Hull; Peter J Brown; Nick S Ryan; Anind K Dey, 1998; Nissanka B Priyantha; Tom Gross; M Seate; Harry Lik Chen).

Based on the proposed context definitions we believe that the most convenient context definition which reflects the mobile environment's requirements is the definition provided by (Gregory D Abowd et al, 1999) which will be used in our research. This definition achieves the highest concept of the contextual information and makes context easily in representation and modeling.

3.2. Context Classifications

The context includes a wide range of contextual information. So, it is necessary to classify a context to certain groups. Many researchers introduced a multiple view of context types. (Schilit et al, 1994) classified the context into three categories, where you are (location context including which physical environment resources are located with the user), who you are with (social context), what (ICT) resources are nearby. (SuanKhai Chong et al, 2007) refers to context as Computing Context, Sensor Context, Physical Context, Time Context and Historical Context. (Wei Liu et al, 2011) proposed a context classification as: Computing Context: available processors, devices accessible for user input and display, nearby resources such as printers, displays, and workstations, network capacity, connectivity, costs of computing and communication, and bandwidth. User's Context: user's location, collection of nearby people, user's profiles and social situation. Physical Context: lighting, temperature, noise and humidity level, traffic conditions. Fig-3 shows the most important context classifications.

Indeed, none of the existing context classifications can satisfy all context-aware application's requirements, because most of these systems relying on a special perspective. These perspectives express the goals of using the context. Also, each classification is appropriate to a specific context-aware system and not generalized to all systems.

In this paper, we will introduce two new context's classification approaches based on two different perspectives as the following.

1) **Context changing degree**: Contextual information can be able to classify according to the degree of context values changing over the time.
2) **Context dependency**: classify the context based on entity's dependency; the contextual information may be user/device dependent or independent.

In the first perspective, the contextual information can be classified according to context changing rate over the time to STATIC, SEMI-STATIC and DYNAMIC context:

- **Static context**: expresses invariant information that does not change over the time, such as user's birth date and age.
- **Semi-static context**: this kind of contextual information changes in fixed intervals or has a limited changing rate. For example, the day time's change over time but these changes are limited to fixed intervals (e.g. morning, afternoon and evening).
- **Dynamic context**: Refers to the contextual information that is frequently changing, the changing in context value may be very slow, slow, medium, fast or very fast. Also, the changing rate depends on a user and a device usage. For any context attribute the contextual information changing rate differs from a user to another and from a device to another. For example, physical environment of an entity, e.g. noise level, air pressure.

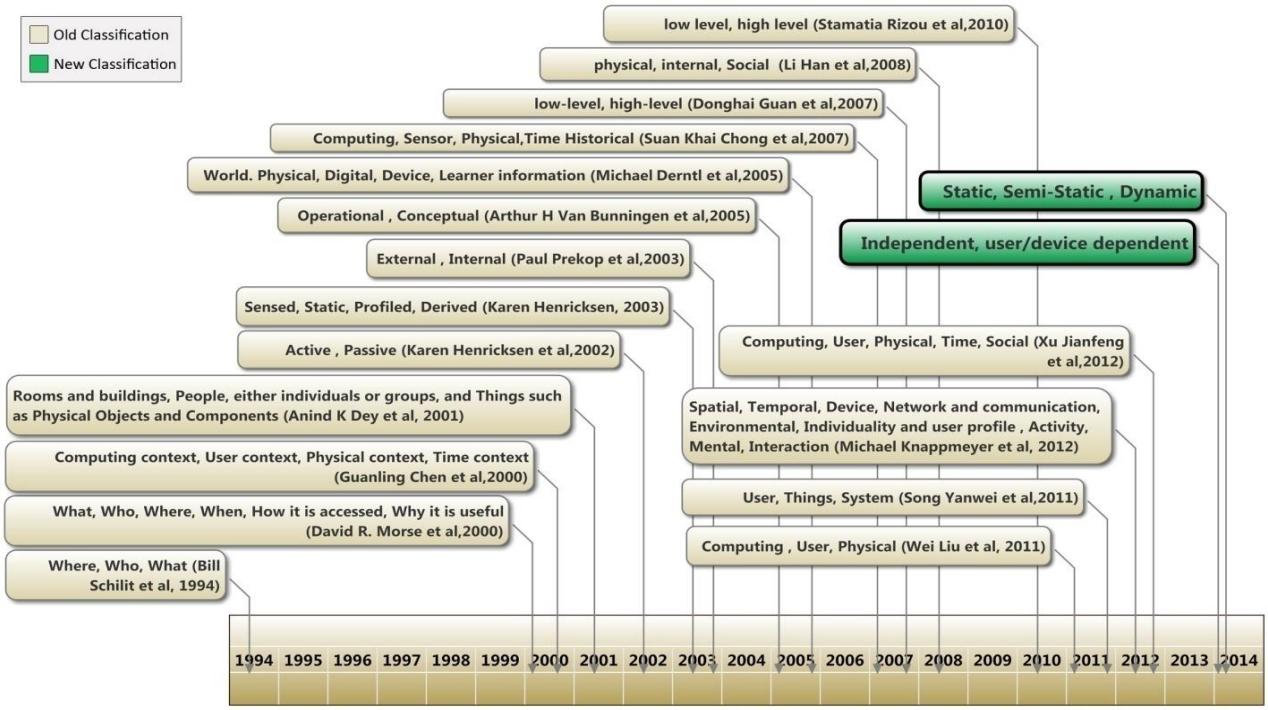

Fig. 3: Context classifications

In the second perspective of context classification, the contextual information can be classified according to the dependency as the following:

- **Independent Context**: The contextual information that does not belong to the device or the user such as environmental temporal context (Information about the absolute time, relative time and day time).
- **User/Device Dependent Context**: The contextual information that characterizes the user or the device such as device's context (processing capabilities, input sensors and visualization capabilities) or user's activity (Information about what an entity does) or user's profile.

4.Context Retrieval Problem

Context retrieval problem is one of the key issues in the context-aware systems. This problem deals with how to retrieve the required context with minimal cost for data management in the mobile environments. In this section, we will propose and formulate this problem.

4.1 Definitions, Assumptions and Models

The main objective of this study is to propose a systematic solution to minimize the cost of context retrieval. We assume that there is a mobile environment which contains a set of mobile users *U*. This environment utilizes the context to operate its applications. The utilized context consists of a set of different context attributes *CA* such as (location, time, and user's activity). So, we have some users who utilize some applications that rely on the context to accomplish its jobs. Each of these applications has its own context attribute query plan and uses this context in a periodical form that differs from other systems. As we have mentioned in section III, the context value changes over time and these changes may be STATIC, SEMI-STATIC, or DYNAMIC. So, the context-aware middleware must support these applications by a reliable and up to date context, which is appropriate to context changing rate. According to this situation the context retrieval problem arises. Such a middleware queries a new context and in other cases some context do not need to be updated. This operation overloads the context-aware system and reduces the total energy. To decrease system's overload, we propose new context retrieval approaches to minimize the energy's consumption from context retrieval in mobile environments. Firstly, we will introduce some basic definitions of terms that will be used in the context retrieval problem formulation.

Definition (1).Context changing time *ct*: Represents a period of time that the context takes to change from a value to another.
Definition (2).Context changing ratio: The ratio of context's value changing over time.
Definition (3).Context changing time probability *P(ct)*: Represents the probability of occurrence a specific context changing time in a limited period of time.
Definition (4).Context expiration date/time: A specific period of time for each context attributes. After elapsing this period the context should renew its value.
Definition (5).Context changing time importance factor *I*: Expresses the sensitivity value of context changing.

The contextual information and context changing rate may obtain directly from user or by observing user activities and device status for a period of time. So, we can classify the context retrieval process into:

- **Reactive retrieval**: This means that the context changing rates are known in advances for all users and can be acquired directly from the user. With the reactive scheme, the user can determine the context changing rate and the context changing time of the context value. Also, the changing rate may be determined by the system, for example, a context-aware system already knows the changing rate of day time (morning, afternoon and evening).
- **Predicted retrieval**: In this type the context changing rate is unknown and will be predicted from the observation of a user and a device in a limited period of time. After that the context-aware system can predict the changing rate and context changing time.

So, our problem now is how to determine these values? To answer this question, firstly we will formulate the context retrieval problem and then introduce our new approaches to solve it.

4.2 Problem Formulation

In this subsection, we will formulate our context retrieval problem. First, we will describe energy consumption model. Second, we will formulate a context retrieval cost. Finally, we will propose our objective function.

- **Energy Consumption Model**: The energy model is concerned with the dynamics of energy consumption of a mobile sensor. We introduce an energy model to retrieve a context from sensor connected to a mobile device as the following: Assume R_{ca} is the consumed energy to retrieve k bits of context attribute $ca \in CA$. Then, the energy consumption to retrieve any context from sensor belong to mobile device is defined as the following:

$$E_{ca} = k_{ca} R_{ca}\,, \quad ca \in CA \tag{1}$$

- **Context Retrieval Total Cost C_t:** The total cost of context retrieval operation C_t will be calculated as follows.

$$C_t = C_n + C_a \tag{2}$$

Where C_n represents context attribute observation cost and C_a is algorithm execution cost.

$$C_n = N E_{ca} + C_c \tag{3}$$

Where N represents the number of contest requests, C_c represents the cost of computing context changing time probability and obtaining final selected context changing time. Value of C_n computed in context attribute observation phase for one time only.

- **The objective Function**: Our objective function is to minimize the total cost C_t of context retrieval process, which can be written using the linear programming approach as follows.

$$Minimize\ C_t = N E_{ca} + C_c + C_a \tag{4}$$

Such that:

$$E_{ca} \geq 0 \quad , \quad C_c \geq 0\,, C_a \geq 0 \tag{5}$$

$$pE_{ca} + q\,(C_c + C_a) \leq M E_{ca} \tag{6}$$

Where

- ***Constraint*** (5) means that sensor's energy consumption from retrieving the context, cost of computing context changing time probability and obtaining final selected context changing time, and algorithm execution cost all of them larger than *0*.
- ***Constraint*** (6) means that the cost of the new context retrieval method (L.H.S) is less than or equal to the cost of the old context retrieval model (R.H.S). Where p is the number of context requests in observation phase, and q is the number of context retrieval in a limited period of time using the new retrieval algorithms, and $p + q = M$.

5. The proposed Context Retrieval Algorithms

5.1 The basic Idea

Our basic idea is based on using the context attribute's changing time to reduce the number of context retrieval times. At the starting point of context retrieval operation, the context-aware system observes the context's value changing, and must describe the observation method for each context attribute, determines the maximum and minimum observation time, finally specifies the number of context's requests. Also, the context-aware system stores context value meanwhile the specified time for each context attribute. After that, the context-aware system can calculate the changing time ratio value *ct* at the ending of the observation period for each context attribute. Then the context-aware system will calculate the context changing time probability $P(ct_i)$, $i=1,2, \ldots ,q$, where q is the number of context changing periods. The context-aware system should also determine the expiration date/time for each changing ratio. After the expiry of the validity period the context-aware system must obtain a new context changing time. The importance factor I of each context attribute *ca* expresses the sensitivity of the changes in context attribute value; I will be between 0 and 1, where 1 represents the highest importance value, the context-aware system will choose the minimum change period for this context attribute. The importance value I can be determined by one of the following two methods:

- Asking user for the context importance value, in case if the context attribute represent user.
- Estimating the importance value depending on the behavior of the context attribute.

As for second method, we emphasize that the best way to obtain the importance factor I is using any computation intelligent methods or decision making methods.

Now, we will formulate the importance factor I using linguistic variables (Very Low, Low , Medium, High, or Very High), which represent the ratio scale that is employed to compare the probability of changing time $P(ct_i)$ according to the linguistic meaning from L to V. such that L,V are minimum and maximum probability of context changing time respectively. We denote to the importance values as follows.

$$\underbrace{L,\ldots,a1}_{\text{VeryLow}}\underbrace{a1,\ldots,a2}_{\text{Low}}\underbrace{a2,\ldots,a3}_{\text{Medium}}\underbrace{a3,\ldots,a4}_{\text{High}}\underbrace{a4,\ldots,V}_{\text{VeryHigh}}$$

Where *a1*, *a2*, *a3*, *a4* are located between *L* and *V*. The value of *I* can be computed from the next equation:

$$I: f(sp, u_I) = W_1\, sp + W_2 u_I \quad (7)$$

Where system's selected probability $sp \in \{ P(ct_i)\}$, $i=1,2,\ldots,q$.

Such that W_1 is weight of system's suggested context changing time probability *sp*, W_2 is weight of user's suggested context changing time probability u_I. The system's user will specify one of the previous linguistic variables (Very Low, Low, Medium, High, or Very High) to represent the sensitivity value of context to the changing in value. Then, the context-aware system will convert this value to the corresponding context attribute changing time probability $P(ct_i)$, For example, if a user specifies the sensitivity value of any context attribute changing to *High* where the corresponding probability interval's value is [*a3* , *a4*] which is defined by the system and assume that we have context attribute changing time probabilities (0.62, 0.13, 0.05, 0.12, 0.06, 0.02). The context-aware system will select the highest context changing probability value that lies between these intervals. Thus, the value of u_I will be the maximum changing time probability between *a3* and *a4*. The context-aware system will commit to determine the values of W_1 and W_2such that:

$$W_1 + W_2 = 1 \quad (8)$$

Now we can derive the selected changing time as follows.

$$st \in \{ ct_i\}\ ,\ i=1,2,\ldots,q \quad (9)$$

Such that:

$$|P(st) - I\ | = min\{\ |\ P(ct_i) - I\ |\ \},\ 1 \le i \le q \quad (10)$$

Where (10) means that the difference value between importance *I* and the selected changing time *st* is the least difference value amongst the other difference values between all changing times and importance *I*.

5.2 The Proposed Algorithms

Minimizing the context retrieval cost in mobile environments based on using the old context values if it is valid instead of retrieving a new context value. In the traditional context retrieval method the context-aware system requests a new context value from the middleware, and then the middleware obtains a new context value from the sensors or any data source. This operation decreases mobile's resources like (energy and computation) because of the context may have the same old value. To reduce context retrieval cost the middleware needs to perform the following steps:

- **Step1**: Observing the context attribute values for a limited period of time and obtaining the context attribute changing time *st*.
- **Step2**: knowing the last retrieving date/time *LR* for each attribute this value can be obtained from context meta date. According to (Michael Knappmeyer et al, 2012) ***Context meta data***: refers to information explicitly attached to the context data. Meta data may be subdivided into mandatory and optional parameters and may comprise the detection time, validity period, source of information, quality of context, probability or uncertainty, associated entity, etc.
- **Step3**: Using the values of *st* and *LR*. We can decide if the context need to updated or not comparing with the current date/time.

The Traditional Retrieval Algorithm *TRA*, is based on retrieving the context when needed from a sensor or a device or any data source. So, we propose two algorithms that follow the previous described steps to reduce context retrieval cost. The first algorithm is called *One Observation Retrieval Algorithm* (*OORA*). OORA observes each context attribute for a period of time and saves the context value and the retrieving time. Then infers the changing in context value, and then computes the context changing ratio as stated in the preceding sections. The context-aware system relies on this value for a long time until the end of the validity of the value as estimated depending on the type of context attribute. After elapsing the validity period the context attribute should renew its changing time ratio. The second algorithm called *Multiple Observation Retrieval Algorithm* (*MORA*). MORA observes the values of context for a period of time then calculates the value of context change time, and then the system uses this value for a short period of time. Then re-calculates a new context change time value depending on the retrieved context values at the last period of using the context as a new observation period. The difference between OORA and MORA algorithms is that OORA calculates the context changing time value for one time only and then uses it for the rest of the validation time, but MORA recalculates the value of the context changing ratio after a specified time period by using the previous time period as a new observation period. In other words, MORA divides the validated interval time into multiple observation periods while OORA uses only the first initial observation period.

Algorithm 1 and Algorithm 2 show the steps of *OORA* and *MORA*, respectively.

Algorithm 1: OORA

```
Input: ContextExpirationTime
st←GetSelectedChangingTime()
LR ←GetLastRetrievalTime()
CurrentTime←GetCurrentTime()

if CurrentTime>= ContextExpirationTime then
st←GetSelectedChangingTime()
end if
if st+ LR ≤CurrentTime then
    if Ct ≥User/DeviceConditions then
        ContextValue←OldContextValue
    else
        ContextValue←RetrieveNewContextValue
    end if
else
    ContextValue←OldContextValue
end if
```

Algorithm 2: MORA

Input*: ObservationPeriod*

```
st←GetSelectedChangingTime()
LR ←GetLastRetrievalTime()
CurrentTime←GetCurrentTime()
LastStTime←GetLastStRetrievalTime()

if CurrentTime-LastStTime≤ObservationPeriod then
    st←GetSelectedChangingTime()
end if
if st+ LR ≤CurrentTime then
    if Ct ≥User/DeviceConditions then
        ContextValue←OldContextValue
    else
        ContextValue←RetrieveNewContextValue
    end if
else
    ContextValue←OldContextValue
end if
```

Where *st* is the selected context changing time, *LR* is the last context retrieval time, and C_t is the total cost of context retrieval for each context attribute.

These two methods will decide if the context-aware system will use the old context value or will retrieve a new context value from the data source based on context changing time *st*. In these methods we take into consideration if the context retrieval cost is very high comparing to the device resources (energy, memory and time) and user's conditions. If so, then the context-aware system can take a decision to use the old context value instead of retrieving a new context value. Fig-4 illustrates the context retrieval algorithm layer.

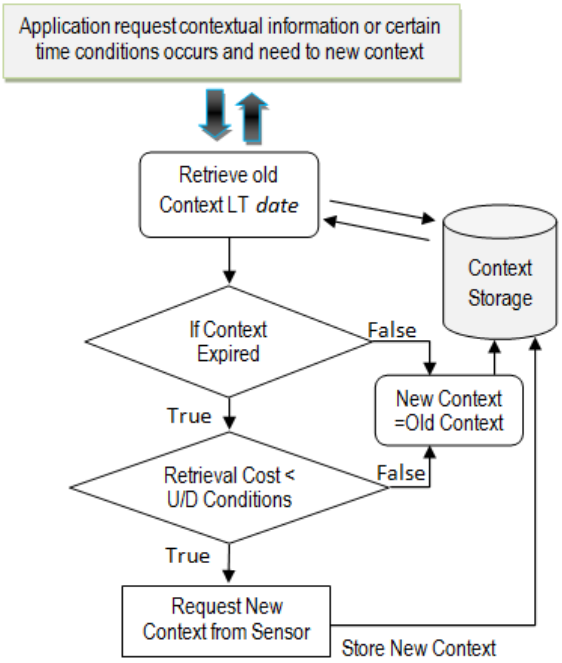

Fig. 4: Context Retrieval Layer

The context-aware system that uses different context types must be supported with the most up to date contextual information in order to adopt with the changes in context values. So, these systems retrieve the contextual information at any time from the sensor or any information resource, which increases the energy consumption. Using our retrieval methods will reduce the energy consumption from context retrieval algorithm, because our algorithms can use some of some old values based on the context changing time probabilities. So, we can introduce the following theorem and its proof as follows:

Theorem 1.*If there are a set of context attributes CA and have M dynamic attributes and N static and semi-static attributes then the total cost of our context retrieval algorithms (OORA and MORA) will be lower than the cost of traditional context retrieval algorithm (TRA) for all values of M.*

Proof: let a context-aware system uses *M* context attributes and the power consumption cost for context retrieval is C_r, if this system requests context value *H* times at a specific period of time, when we use the traditional context retrieval method the total power consumption cost equal:

$$\sum_{i=1}^{M} H_i C_r \tag{11}$$

In case of using our new context retrieval methods the total number of requests are divide to p_i which represents the number of requests that need to retrieve a new context value and q_i which represents the number of requests that use the old context values (without retrieving new value) for each context attribute *i*, the total power consumption cost by using our retrieval methods is:

$$\sum_{i=1}^{M} p_i C_r + q_i C_a \tag{12}$$

Where, C_a is cost of using old context value. It is clear that retrieving a new context from a sensor or any information resource is larger than retrieving the old context value. So, we can conclude that $C_r > C_a$. Therefore, there is a difference between using the traditional method and using our new methods whereas q_i ever larger than *0*. So, we can say that the cost of using our new retrieval methods is lower than the cost of using the traditional context retrieval method.

6. Simulation Results

In this section, we present the simulation results of a simple scenario of context retrieval on a mobile node. This scenario describes a context-aware middleware that retrieves a context from a sensor or any other information resource. The context-aware middleware will support all context-aware applications by all their needs of the contextual information.

The simulation will study the performance of the three context retrieval methods: TRA, OORA, and MORA for retrieving a context from a mobile node.

We have simulated this scenario at 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 minutes. Every time we run the simulation, we determine the power consumption from using the three methods where the resulted value will be the average of five simulation experiments at each period of the previous time. So, we obtain three values of the energy consumption cost at each one of the 10 times. We

have implemented our simulation environment using C++ coding in Eclipse platform.

We set system weight W_1 = 0.6 and user weight W_2 = 0.4. Also, we set number of Bits in context k = 64, energy consumption for a bit R = 3.2 *kj*, and Context Observation period = 5 *minutes*.

6.1 Simulation Results and Analysis

In this section, we will discuss the simulation results and compare the energy consumption and efficiency ratio of context retrieval methods TRA, OORA and MORA.

Fig-5 and Fig-6: Show the energy consumption versus time and number of requests (in 60 minutes), respectively for the three retrieval methods. As shown in *Fig-5*, the energy consumption for the three methods increases as time increases, because the number of context requests increases over time, which increases the energy consumption. Also, this figure shows that the energy consumed from using OORA and MORA methods is less than TRA method, because OORA and MORA take into account the stability in context value over the time by computing the context changing time ratio to decide if the middleware will retrieve a new context value or use the old context value. Where, the energy consumed by retrieving the new context value is larger than the energy consumed from using the old context value. As a result, OORA and MORA can use old context values more than TRA do. So, OORA and MORA save energy by using some of the old context values instead of retrieving a new value. As shown in *Fig-6*. Energy consumption for the three methods increases as number of requests increases, because when the number of requests increases, the device will consume additional energy for retrieving new or old context values. Also, this figure shows that the energy consumed by using OORA and MORA methods is less than TRA method because the two new methods take into account the stability in context value over time by computing the context changing time ratio to decide if the middleware will retrieve a new context value or use the old context value.

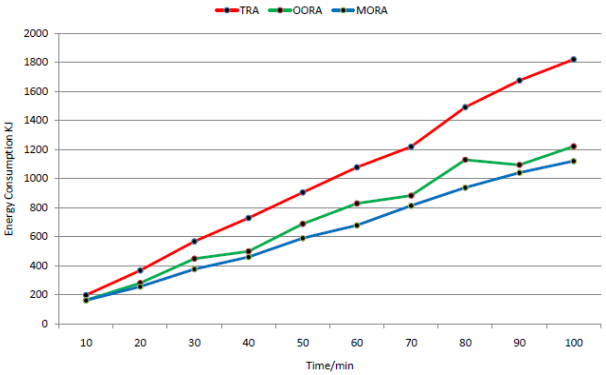

Fig. 5: Energy Consumption vs. Time

As shown in Fig-5 and Fig-6, our retrieval methods, OORA and MORA, can reduce the consumed energy by *26%* and *33%*, respectively, lower than TRA. Also, MORA is better than OORA because OORA computes the context changing ratio for one time at the start of validation time interval and relies on this value for the rest of time. On the other hand, MORA recalculates the value of the context changing ratio after a specified time period by using the previous time period as a new observation period, where MORA divides the validation interval time into multiple observation periods. This technique increases the using of the old context which will reduce the consumed energy for retrieving a context.

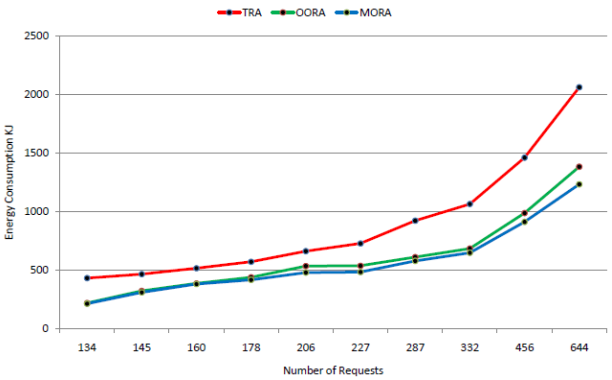

Fig. 6: Energy Consumption vs. Number of Requests

Fig-7 and Fig-8: Show the efficiency ratio versus time and number of requests (in 60 minutes), respectively, where the efficiency ratio represents the proportion of number of times using the old contextual information to the total number of context requests for each context attribute i as follows.

$$Efficiency\ Ratio = q_i/H \quad (13)$$

Fig. 7: Efficiency Ratio vs. Time

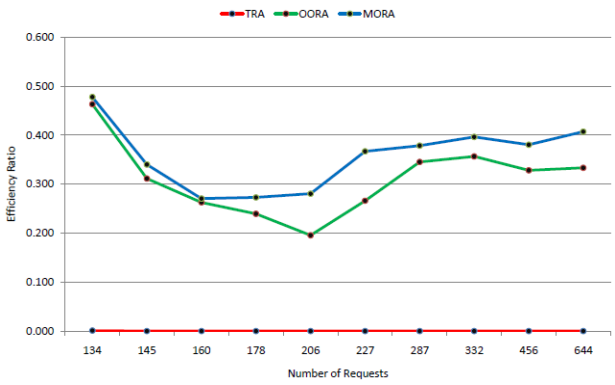

Fig. 8: Efficiency Ratio vs. Number of Requests

As shown in above two figures, the efficiency ratio of OORA and MORA methods is much larger than TRA method, because TRA method never uses the old context values while OORA and MORA methods use some of the old context values. Also, we note that the efficiency ratio of MORA is better than OORA because MORA uses the old context more than OORA method due to the changing of *st* value over the time.

Conclusion

In this paper, we have introduced the most relevant context definitions and context classifications. Also, we have proposed two new context classifications approaches based on the context changing and context dependency. Also, we have presented a comprehensive review and analysis of how context-aware system and middleware systems developed, In addition, we proposed two new context retrieval approaches based on context changing ratio to minimize context retrieval cost called One Observation Retrieval Algorithm, OORA, and Multiple Observation Retrieval Algorithm, MORA. These two retrieval methods can reduce the energy consumption by *26%* and *33%*, respectively, compared to using the traditional context retrieval method, TRA. The simulation results have shown that the efficiency ratio of MORA method is better than OORA method and the efficiency ratio of TRA method ever equals 0.

Our future work involves using the new retrieval approaches to develop a new mobile ranking algorithm based on context to enhance the routing protocols in the mobile networks.

References

Peizhao Hu, Jadwiga Indulska, and Ricky Robinson (2008).An autonomiccontext management system for pervasive computing.*InPervasiveComputing and Communications, 2008.PerCom 2008. Sixth AnnualIEEE International Conference on*, pages 213–223. IEEE.

Peter Brown, Winslow Burleson, Mik Lamming, Odd-WikingRahlff,Guy Romano, Jean Scholtz, and Dave Snowdon (2000). Context-awareness:some compelling applications. *In Proceedings the CH12000 WorkshoponThe What, Who, Where, When, Why and How of Context-Awareness.*

Matthias Baldauf, SchahramDustdar, and Florian Rosenberg (2007).A surveyon context-aware systems.*International Journal of Ad Hoc andUbiquitous Computing*, 2(4):263–277.

Stefan Poslad (2009). Ubiquitous computing: smart devices, environments andinteractions. *Wiley.com*.

FaustoGiunchiglia (1993).Contextual reasoning.Epistemologia, special issueon I Linguaggi e le Macchine, 16:345–364.

Bill Schilit, Norman Adams, and Roy Want (1994). Context-aware computingapplications.*In Mobile Computing Systems and Applications, 1994.WMCSA 1994.First Workshop on*, pages 85–90.IEEE.

Andy Ward, Alan Jones, and Andy Hopper (1997).A new location techniquefor the active office.*Personal Communications, IEEE*, 4(5):42–47.

David Franklin and Joshua Flaschbart (1998). All gadget and no representationmakes jack a dull environment. *In Proceedings of the AAAI 1998 SpringSymposium on Intelligent Environments*, pages 155–160.

Jason Pascoe (1998). Adding generic contextual capabilities to wearablecomputers.*In Wearable Computers, 1998. Digest of Papers. SecondInternational Symposium on*, pages 92–99. IEEE.

Tom Rodden, Keith Cheverst, K Davies, and Alan Dix (1998).Exploitingcontext in hci design for mobile systems.*In Workshop on humancomputer interaction with mobile devices*, pages 21–22. Citeseer.

Gregory D Abowd, Anind K Dey, Peter J Brown, Nigel Davies, MarkSmith, and Pete Steggles (1999).Towards a better understanding of contextand context-awareness.*In Handheld and ubiquitous computing*, pages304–307.Springer.

Anind K Dey (2001). Understanding and using context.Personal andubiquitous computing, 5(1):4–7.

Henry Lieberman and Ted Selker (2000).Out of context: Computer systemsthat adapt to, and learn from, context.*IBM Systems Journal*,39(3.4):617–632.

Guanling Chen, David Kotz, et al (2000). A survey of context-aware mobilecomputing research.*Technical report, Technical Report TR2000-381,Dept. of Computer Science, Dartmouth College*.

Gregory D Abowd and Elizabeth D Mynatt (2000). Charting past, present,and future research in ubiquitous computing. *ACM Transactions onComputer-Human Interaction (TOCHI)*, 7(1):29–58.

Karen Henricksen (2003). A framework for context-aware pervasive computingapplications.*University of Queensland*.

Paul Dourish (2004). What we talk about when we talk about context. *Personaland ubiquitous computing*, 8(1):19–30.

Ling Feng, Peter MG Apers, and Willem Jonker (2004).Towards contextawaredata management for ambient intelligence.*In Database andExpert Systems Applications*, pages 422–431. Springer.

SungjinAhn and Daeyoung Kim (2006).Proactive context-aware sensornetworks.*In Wireless Sensor Networks*, pages 38–53. Springer.

Andreas Zimmermann (2007). Context Management and Personalisation:A Tool Suite for Context-and User-Aware Computing. PhD thesis,Universit¨atsbibliothek.

C. Bolchini, C. A. Curino, G. Orsi, E. Quintarelli, R. Rossato, F. A.Schreiber, and L. Tanca (2009). And what can context do for data? Commun.*ACM*, 52(11):136–140, November.

Constantin Schmidt (2011). Context-aware computing.Berlin Inst. *Technologytech.rep*.

Michael Knappmeyer, S Kiani, EikeReetz, Nigel Baker, and RalfTonjes (2012).Survey of context provisioning middleware.

CharithPerera, ArkadyZaslavsky, Peter Christen, and DimitriosGeorgakopoulos.Context aware computing for the internet of things: Asurvey. *IEEE*, 2013.

Bill N Schilit and Marvin M Theimer (1994).Disseminating active mapinformation to mobile hosts.Network, *IEEE*, 8(5):22–32.

Randy H Katz (1994). Adaptation and mobility in wireless informationsystems.*Personal Communications, IEEE*, 1(1):6–17.

CE Billings (1995). Situation awareness measurement and analysis: A commentary.*In Proceedings of the International Conference on ExperimentalAnalysis and Measurement of Situation Awareness*, pages 1–6.Daytona Beach, FL: Embry-Riddle Aeronautical University Press.

Richard Hull, Philip Neaves, and James Bedford-Roberts (1997).Towardssituated computing.In Wearable Computers, 1997.Digest of Papers.*First International Symposium on*, pages 146–153.IEEE.

Peter J Brown, John D Bovey, and Xian Chen (1997). Context-aware applications:from the laboratory to the marketplace. *Personal Communications,IEEE*,4(5):58–64.

Nick S Ryan, Jason Pascoe, and David R Morse (1998). Enhanced realityfieldwork: the context-aware archaeological assistant. *In Computerapplications in archaeology*. *Tempus Reparatum.*

Anind K Dey (1998). Context-aware computing: The cyberdeskproject.*In Proceedings of the AAAI-Spring Symposium on IntelligentEnvironments*, pages 51–54.

Nissanka B Priyantha, AnitChakraborty, and HariBalakrishnan (2000).Thecricket location-support system.*In Proceedings of the*

6th annualinternational conference on Mobile computing and networking, pages32–43.ACM.

Tom Gross and Marcus Specht (2001).Awareness in context-aware informationsystems.*In Mensch & Computer 2001,* pages 173–182. Springer.

M Seate and B Dwolatzky (2002).Adaptive context human-computer interface(hci) for power distribution network maintenance teams.*In AfriconConference in Africa, 2002. IEEE AFRICON. 6th*, volume 2, pages907–912.IEEE.

Harry Lik Chen (2004). An intelligent broker architecture for pervasive contextawaresystems. *PhD thesis, University of Maryland, Baltimore County*.

SuanKhai Chong, Ian Mccauley, SengWaiLoke, and ShonaliKrishnaswamy (2007).Context-aware sensors and data muling.*Citeseer*.

Wei Liu, Xue Li, and Daoli Huang (2011).A survey on context awareness.*In Computer Science and Service System (CSSS), 2011 InternationalConference on*, pages 144–147. IEEE.

David R. Morse, Stephen Armstrong, and Anind K. Dey (2000).The what,who, where, when, and how of context awareness.

Anind K Dey, Gregory D Abowd, and Daniel Salber (2001).A conceptualframework and a toolkit for supporting the rapid prototyping of context-awareapplications.*Human-computer interaction*, 16(2):97–166.

Henricksen, Jadwiga Indulska, and AndryRakotonirainy (2002).Modelingcontext information in pervasive computing systems.*In PervasiveComputing*, pages 167–180. Springer.

Paul Prekop and Mark Burnett (2003).Activities, context and ubiquitouscomputing.*Computer Communications*, 26(11):1168–1176.

Arthur H Van Bunningen, Ling Feng, and Peter MG Apers (2005).Context forubiquitous data management.*In Ubiquitous Data Management, 2005.UDM 2005. International Workshop on*, pages 17–24. IEEE.

Michael Derntl and Karin Anna Hummel (2005).Modeling context-awaree-learning scenarios.*In Pervasive Computing and CommunicationsWorkshops, 2005.PerCom 2005 Workshops. Third IEEE InternationalConference on*, pages 337–342. IEEE.

Donghai Guan, Weiwei Yuan, Sungyoung Lee, and Young-Koo Lee (2007).Context selection and reasoning in ubiquitous computing. *In IntelligentPervasive Computing, 2007.IPC. The 2007 International Conferenceon*, pages 184–187. IEEE.

Li Han, SalomaaJyri, Jian Ma, and Kuifei Yu (2008).Research on contextawaremobile computing.*In Advanced Information Networking andApplications-Workshops, 2008.AINAW 2008. 22nd International Conferenceon*, pages 24–30. IEEE.

StamatiaRizou, Kai Haussermann, Frank Durr, NazarioCipriani, andKurtRothermel (2010).A system for distributed context reasoning.*InAutonomic and Autonomous Systems (ICAS), 2010 Sixth InternationalConference on*, pages 84–89.IEEE.

Song Yanwei, Zeng Guangzhou, and PuHaitao (2011).Research on the contextmodel of intelligent interaction system in the internet of things.*In ITin Medicine and Education (ITME), 2011 International Symposium on,volume 2*, pages 379–382. IEEE.

XuJianfeng and Wang Dong (2012).Object-oriented and ontology context-awaremodeling based on xml.*IEEE*, pages 1795 – 1800.