International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-1, November 2019

Contextualization using Context-Aware Publish and Subscribe (CAPS) based on IoT

Deeba K, RA.K. Saravanaguru

Abstract: The Internet of Things (IoT) activates massive data flow in the real world. Each computer can presently be linked to the internet and supply useful decision-making information. Virtually sensors are implemented in every aspect of life. From different sources of sensors can produce raw data. Due to the various data sources, the method of extracting information from the flow of data is mostly complicated, networks inadequate and criteria for real-time processing. In addition, an issue of context-aware data processing and architecture also present, despite the fact that they are essential criteria for stronger IoT structure. In order to meet this issue, we recommend a Context-aware Internet of Things Middleware (CAIM) architecture. This enables the incorporation of highly diverse IoT application context information by using light weigh protocol MQTT (Message Queue Telemetry Transport) for transmitting basic data streams from sensors to middleware and applications. In this paper, we propose a contextualization which means that obtain data from sensors of different sources. First have to create a context profile with the help of context type like user, activity, physical, and environment context. Then also is create a profile by using attributes. Finally, raw data can be change into contextualized data through CAPS (context-aware Publish-Subscribe) hybrid approach. This paper discusses the current context analysis strategies that use either rational models or probabilistic methods exclusively. The evaluation of identifying contextualization methods shows the shortcomings of IoT sensor data processing as well as offers alternative ways of identifying the context

Keywords: CAIM, CAPS, Context acquisition, Context-aware, Contextualization, IoT.

I. INTRODUCTION

The word context-awareness has become commonly used for systems that monitor their environment (Schilit et al., 1994) [1]. Context-awareness is an important portion of ubiquitous computing in which devices are used to offer users with personalized services to suit the environment of the user. Ubiquitous computing incorporates the processing of data into daily objects and operations, enabling machines to adapt to user environments rather than compel users to adapt to computers. As a solution, ubiquitous computing meanwhile undertakes the systems and computational devices for most of the normal activities (Perera et al., 2014) [2].

Revised Manuscript Received on November 30, 2019.
* Correspondence Author
Deeba K, School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India. Email: deebakanmani@gmail.com
RA.K. Saravanaguru*, School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India. Email: saravanank@vit.ac.in

Contextual awareness is the capacity of computer systems to obtain, retrieve, and justify the context and adapt consequently to their apps. A context-aware system begins collecting raw, low-level contextual information, interpreting raw contextual information into a high-level interpreted context, explaining the context interpreted to derive consequences, and adapting the application conduct based on the consequences. Context data can be collected from a wide range of sensors and non-sensor resources, that are used to enable to acquiring an environment's present, often unclear, erroneous and imprecise status (Bettini et al., 2010; Perera et al., 2014 [2,3]).

Context can be classified into three levels: internal context, external context, and boundary context. These context groups into six classes of context, they are

User: it is an internal context and information about the user profile, location, and preferences.

Activity: External context which is used to explains the various set of activities this could be prepared by the context.

Time: It is an external context that says about the data and time of the context. It is used to specify the historic situations in the context

Device: It is an external context which is used to define the software and hardware device by the context.

Location: It is an external context, describes the location of the context.

The Internet of Things (IoT) was invented in the early 1980s and it is the further state of internet development and the internet has gone through several stages. At present, it is progressing about a stage in which all the objects beside us are linked to the internet and can interact with among ourselves. The Internet of Things (IoT) is a novel framework that quickly achieves ground in the conventional wireless telecommunications situation. The fundamental concept of this theory is the pervasiveness of a kind of things and items around us – including tags, detectors, actuators, mobile phones, etc. Which can communicate with one another and collaborate with their community to achieve ordinary purposes through different approaches [5]? The connection of sensors and actuators throughout a network is not entirely novel to the profession of computer science and engineering from that point of view. Over many years, sensor networks were used and investigated [6] Kevin Ashton first coined the phrase ' Internet of Things ' in a 1998 presentation [7]. "The Internet of Things makes it possible to connect people and things to any time, Anyplace, Anything and Everyone, preferably using any path/network and Any service." [8].

II. RELATED WORK

Context information is data on all items (i.e. individuals, locations or things) which are essential to a specific IoT advantage that can be used to contextualize IoT information. One of the most public issues for context-aware intelligent and expert systems is how to acquire contextual information in the IoT environment, where numerous data can be collected from multiple sources in the domain application context, possibilities are rising quickly and context awareness is becoming central[2,9].

The use of sensors is growing, leading to increased competition for information storage systems for sensors. Because of growing scalability and accessibility specifications, NoSQL DBs have acquired impulse in the last few decades. In earlier research, two open-source NoSQL DBs (Cassandra and MongoDB) and one open-source SQL DB (PostgreSQL) were related [10]. The sensor data was collected from electrodes connecting to only the body by a glucose monitoring device. In addition, and also the sensor node's low-power service, the interaction between both the sensor node and the smartphone involves wireless technology to be used. BLE is the optimum result to meet these requirements [11]. Among women who also have a record of miscarriage as demonstrated in as research at the Indian diabetic clinic, there was also an improvement in the rate of gestational diabetes. In cases where GDM formed, an abnormal menstrual cycle has also been noticed to be more frequent [12]. This research showed that, as found in certain studies, obe and overweight women are more likely to develop GDM [13, 14]. The current research found that in the present pregnancy women with such a record of GDM were more probable to have GDM, indicating the inherent tendency of women to acquire insulin indifference [15]. The risk factor of women's diabetes is may differ based on age, family history, BMI and Gender [16]. Later in pregnancy, glucose control affects fetus development, and postprandial normal glycemia can decrease the macrosomia level to less than 10%.[17]. diabetes is a serious illness affected by insulin production(T1D) or insulin secretion and activity (T2D). The BG rate reaches a healthy range of 70-180 mg / dL for people with diabetes [18].

The summary of this article is findings apply to two major directions. We first propose an contextualization based on Context-Aware Publish and Subscribe (CAPS) approach. Secondly, we will implement the contextualization CAPS approach into the healthcare domain in Gestational Diabetic Mellitus (GDM) as a case study.

The paper's description is as pursues. Section 1 explains the Introduction about the context-aware and Internet of Things (IoT). In Section 2 discussed Related works. In section 3 explains the proposed approaches of context-aware fundamentals, layered architecture of contextualization and process of contextualization. Section 4 describes the necessary elements of implementing contextualization with the CAIM architecture.5 presents implementing the scenario in health care domain as a case study. Section 6 presents the implementation and evaluation. Section 7 ends the article and offers some observations.

III. RPROPOSED WORK

Our objective here is to explain how the low-level context can be transformed into a high-level context that will be implemented through an IoT framework. We analyzed numerous context-aware existing works and illustrated their pieces from the view of IoT. The goal here is to consider the process of creating a context that involves some major issues. Here we have discussed about context and context modeling and reasoning types in general.

A. Context

Dey and Abowd given the definition of context in [4] "Context is any information that can be used to characterize the situation of an entities (i.e., whether a person, place or object) that are considered relevant to the interaction between user and an application, including the user and the application themselves."

B. Context-Aware

The author schilit and theimer used the word 'context-aware' in 1994 for the first time [1]. Later, context-aware defined by Dey, "A context-aware system if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task [4]". The configuration and execution can be described in the context-aware system. The particular scenario of the system behavior or actions can be related by the execution.

C. Context Life cycle

Context life cycle exhibits in software systems how the data can be transferred from one stage to another stage (e.g. middleware, application). The context life cycle describes where and where information is being guarded and produced.

D. Context acquisition

Context can be acquired from various types of sensors based on the frequency, responsible and sources.

E. Context modelling

It is necessary to model and describe the information gathered in a significant way. Based on the [perera et al -2014] [2] context modeling has six techniques.

1. Key-value modeling: in this key-value modeling of context information can be in the format of binary file or text files. Compare to other techniques this model described in a simple way. They are simple to handle when lower quantities of information are available.
2. Markup Scheme modeling: Markup scheme models using the tags for data. That is, context will be processed in tags. This method is an enhancement of the modeling method of the key-value model. It enables information to be collected efficiently.
3. Graphical modeling: It is easy to use, learn and well-known modeling. Graphical modeling makes it possible to detect connections in this context model. Hence it is possible to use graphical modeling methods as constant context storage.

International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-1, November 2019

4. Object-based modeling: Object-based model is using class hierarchies and interactions to design data. The object-oriented paradigm towards promoting the re-usability and encapsulation.
5. The logic-based model to portray information regarding context is through expressions, rules, and
6. facts. In addition, to create depictions based on rule or logic, extremely advanced and interactive graphical methods can be used. Logic-based modeling to enable high-level context data which will be extracted from the low-level context. Hence, by behaving as a compliment, it has the ability to improve other context modeling methods.
7. Ontology-based modeling: semantic techniques are used to arrange the context into ontologies. Based on the required, various numbers of standards (RDF, RDFS, and OWL) can be used. It enables more expressive context representation and supports semantic reasoning.

F. Context Reasoning

The reasoning is the method of creating a hypothesis or inference using various logical approaches from the appropriate knowledge or information. While gathering raw data from sensors, it must be transformed into useful information that is needed for the context-aware process. When background information is gathered from various sources, information may be vague. Incomplete sensors can also generate erroneous data. Therefore reasoning enhances the quality of the perceived data that is unclear, incomplete, incorrect and unknown.

1. Case-based reasoning: Case-based reasoning is to obtain instances of learning for the first time. Then we will tag them following the awaited effects. Then we extract a feature that uses the training data to achieve the desired solutions. This approach is commonly utilized in the sensing of cell phones [19].
2. Rule-based reasoning: Rule-based reasoning is a technique to exploit the rule-based information, among all of them, these are the easiest and most direct reasoning technique. By using the low-level context, it enables to produce high-level context knowledge. Rules are especially in the form of IF-THEN-ELSE [20].
3. Fuzzy logic: Instead of defined and precise reasoning, it enables estimated reasoning. Fuzzy logic is comparable to probabilistic reasoning, but trust values instead of probability depict membership degrees. Fuzzy specify the value of the range is [0.0, 0.1] which is represented in [21].
4. Probabilistic logic: This classification of approaches makes it possible to make decisions based on the probabilities linked to the problem-related facts. This can be used to integrate two separate sources of sensor data [22].
5. Bayesian networks: Bayesian Networks are a method focused on abstract probabilism. To illustrate events and relationships between them, it utilizes directed acyclic graphs. It is a method that is commonly used in statistical reasoning [22].
6. Ontology Reasoning: This is built on a logic of explanation, which is a family of formal representations of knowledge based on logic. Ontological reasoning is endorsed primarily by two ordinary web language representations: RDF(S) and OWL [23,24] and Finally the layered architecture represented in figure 1.

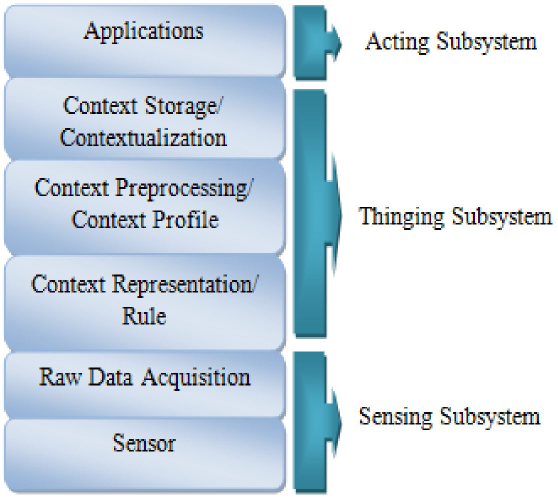

Fig. 1. Contextualization Layered Architecture

G. Process of Contextualization

The diagram shows the context-aware process working process in fig 2. In the real-time situation, how different sensors catch the information and then further procedures such as data fusion, information processing, and context inference happen. It is introduced to different application fields since having a model prepared for context-aware systems. Some of the applications are robotics, traffic monitoring, medical applications, agriculture, etc.

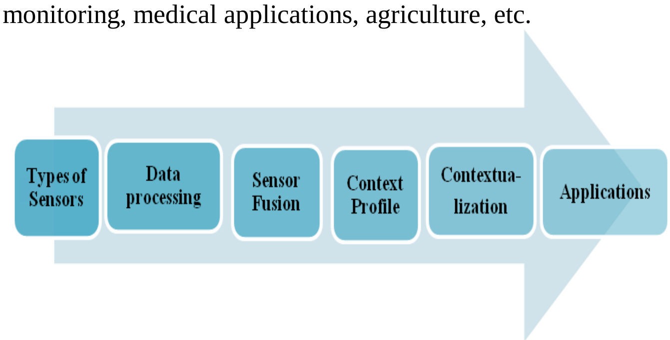

Fig. 2. Process of Contextualization

The context-aware method implies a sequence of subprograms such as context acquisition, context processing (context Representation), and use of context. Data such as stress, temperature, etc. might be recorded and thus sent for enhancement in context acquisition with the assistance of sensors. Some other operations are engaged in the next phase of context processing, such as noise removal (Use of filters to remove existing sensor data and noise), Data calibration (The purpose of data calibration is to update or correct a machine) context interpretation (It is about to analyze and understand the contextual information) and context prediction(It is generally a method of predication on previous estimates).

Overview of the CAIM framework services is represented in table 1.

Table I: CAIM Services

ID	Available Services	Description
1	Context Acquisition	The context must be obtained from different sources. Available services are filtering, aggregation of the context.
2	Context Modeling	It enables structured data with a more significant correlation among data.
3	Context Reasoning	It is a method of inferring new knowledge. On the basis of possible context, it has a clearer understanding.
4	Context Distribution	It's quite a direct task. It offers techniques to the user with context.
5	Context Profile	Services in charge of the context entity, context query and context request.

IV. SYSTEM ARCHITECTURE

A. Entity

An entity is a physical or virtual object that can be associated with one too many context attributes which can be any type of data that characterizes this entity.

B. Context Service

Context service provides contextual information about a particular entity. Context service represented as a triple or tuple (E, C, P) here E is an entity, C is context attributes, and P means predicate.
Based on the user's profile, that is a calendar and social profiles understand where they are and what they're doing and can find out who is working presently.
Example: Entity: Diabetic patient
Attributes: Height, Weight, BMI, Glucose.

C. Context Query

It is requested for contextual information extracted from one or more entities (HLC).

D. Context Request

A request for contextual information about a particular entity.

Fig 3 represents the overall Context-aware IoT Middleware framework in this contextualization process explained in detail.

If we are unable to properly identify the present context, we will need to derive from the profiles of the beginner to obtain contextual data related to the request. It is used to get the information from the beginner's profile's preferences through the run time. Here, two types of methodologies to extract the context, they are, one is default context is to be extracted from the user profile and the preferences. It's quite another to obtain from either the social profile, location profile, or calendar profile derived contextual data.

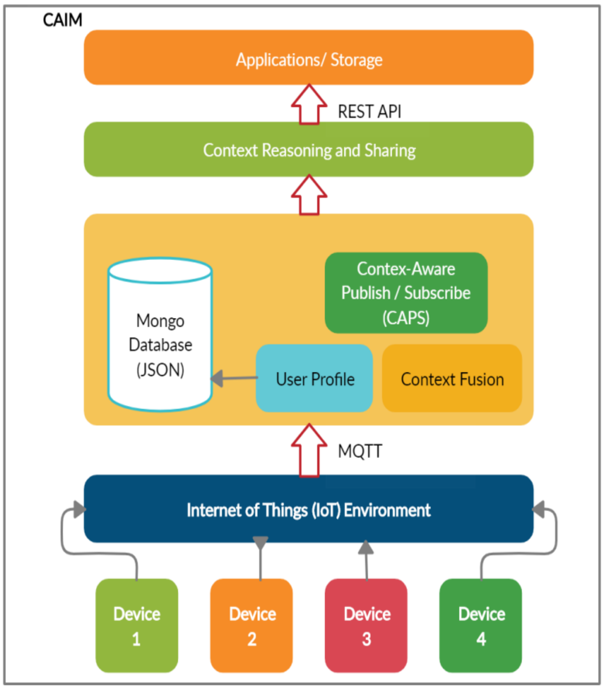

Fig. 3. Architecture of contextualization with CAIM

First, the user should determine their values in the desired characteristics from this method such as name, ID, email, etc. because several properties identified in the user profile and outcome of preference have standard values. If the property values are not specified clearly then our system fills the user's default value.

Secondly, we obtain contextual data from a calendar and social profiles in the second method. Here to create a user profile, we will take an example of a social profile and calendar profile, which is used to discover the relevant information about the user and it is not mentioned who is working under the social profile. Like users can revolve his/her profile to a private mode so we could not find any matter. So this is what done by the calendar profile and querying by every user.

The following parameters are considered to discuss the functionality
Ui = User Preference with value i.
Ci = Context information with value i.
Ct = Context type
Ei_Ct = Event information with value i for the Context Type
Ct. E_c= Environment Context.
D_c= Device Context
A_c = Activity Context

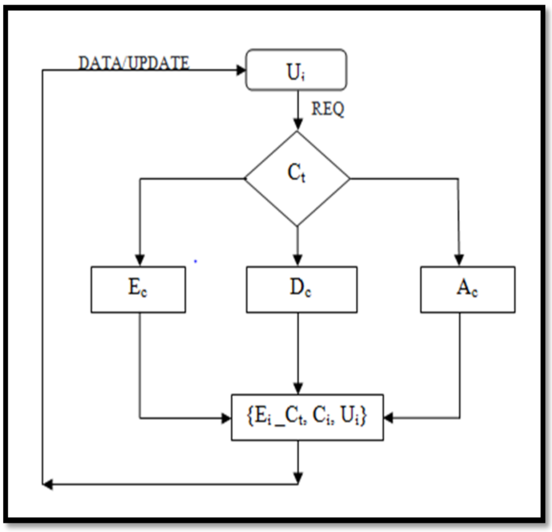

Fig. 4. Sequence Diagram

The flow sequence of the functionality diagram is illustrated in Fig 4. The service consumer task is to provide the request to the service provider by exchanging the user preference and all the related information for the present situation. To get the preferred action from the service provider, the Negotiator is responsible for gathering event information and perform the negotiation process. The negotiation process is a set of rules applied for available services for the current activity. This decides the service priority from the available service set. Also, the negotiation helps to identify the preferred action to be taken by the consumer with the maximum threshold time.

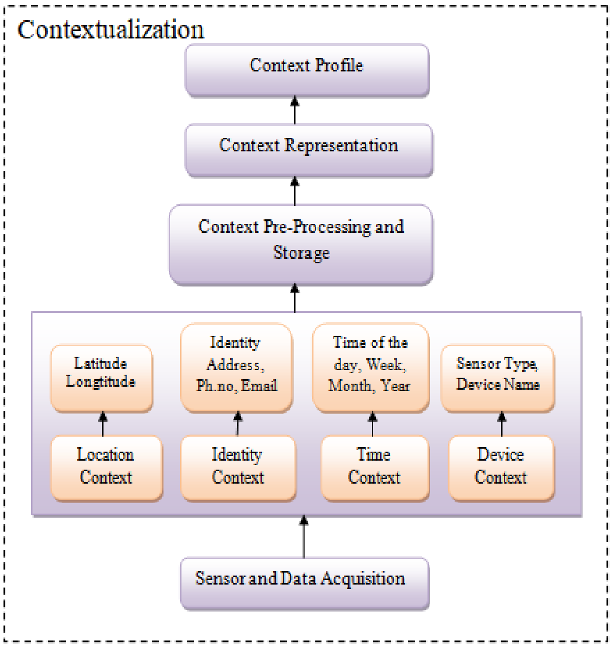

Fig 5: Flow Diagram for contextualization

The diagram shows the contextualization process of context-aware acquisition as represented in fig 5. Various sources of sensors can be produced the data for data representation. Here we have taken four types of context such as Location, Time, Identity, and device context based on these context data that could be pre-processed and stored. Then the pre-processed data will be represented for context profile use. Context profile can be created with the help of the JSON file format and stored it into a database for future use. A user details we can update in later. Contextualization means changing the raw sensor data into high-level data or meaningful data this process is represented in figure 5.

V. CASE STUDY - Gestational Diabetic System Design

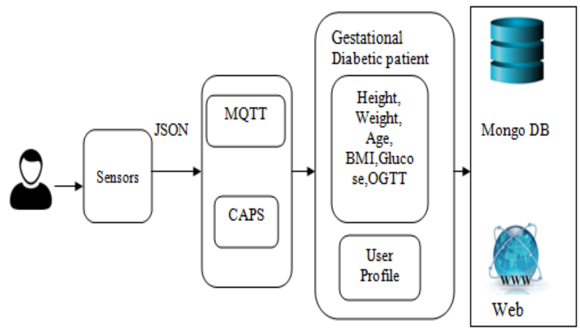

Fig. 6. System design for Gestational Diabetic Patients

Present Gestational diabetic patient information is provided in the software through sensor devices including weight, BG level, and blood pressure, OGTT (Oral Glucose Tolerance Test). In contrast, the selected patient will also be provided with personal information like date of birth, height, and age. This also presents several healthcare suggestions for the current patient to improve his/her quality of health. The patient is suggested to maintain a healthy diet, lose weight, and exercise regularly to improve his/her health quality.

From the sensor, the Gestational diabetic patient information can be obtained through the CAPS and MQTT approach and the data could be represented in the form of the JSON format. Then based on the data, a user profile can be created with the help of rule-based modeling technique, later it can be stored into the Mongo DB or Web. CAPS produces large amounts of sensor data and is produced on an ongoing basis. The NoSQL MongoDB was examined as the DB in this, to store the patients ' sensor data. A simple user (context) profile and context rule can be represented as follows.

A. Rule Generation

After gathering the technical standards from the sources of knowledge, we then translate them into rules that machines can understand such that patients can spontaneously receive based on reasoning regarding their individual context, rule and user profile. Every rule consists of a body and ahead. Rules in this format can be described as phrases “IF-THEN” that adopt the THEN format of the IF.

Rules for this research were developed with the help of domain experts

- If ((Over weight=true)) Then type 2 or pre-diabetes or gestational diabetes.
- If ((Family history of diabetes during pregnancy=true) and (Previous pregnancy=true) and (Baby over 4 Kg during the previous pregnancy=true) and (Sleep walking=true)Then gestational diabetes.

- If ((Vaginal mycotic infection=true)) Then type 1 or type 2 or gestational diabetes.
- If ((Increase thirst=true) and (Increase urge to urinate=true) Then type 2 or pre diabetes.

B. User Profile


```
Var PatientData =
 {
   "Patient_id" : "12345"
                  " Personal":   {
                        "Name"           : "Anu"
                        "Admitted Date": "15/07/2018"
            "Gender"           : "Male"
            "Age"              : "31"
                        "Address"        : {
      "Street": "1/20 Anna street"
      "City" : Vellore
      "State": "Tamilnadu"
                              }
                                      },
              "Profile": {
          "Patient _id"           :"12345"
          "Weight"                : "70Kg"
          "Diabetic Parent"      : "Yes"
          "Nausea"                : "No"
          "Sweating"             :"No"
          "Frequent Urination":"Yes"
          "Status"               : "Need to check all tests"

   "Test Details" : {
          "Patient_id":   "12345"
         "Age"          : "31"
    "Weight"      :"70"
    "BMI"          : ">25.0"
    "Initial Glucose Test": ">140 mg/dL
    "OGTT": "High glucose repeated 2  more tests"
    "Sugar in Urine": "Yes"

                      }
        }
```


C. Context-Aware Publish-Subscribe (CAPS)

MQTT (Message Queuing Telemetry Transport) is a protocol running on part of the TCP / IP (Transmission Control Protocol / Internet Protocol) layer. Implementing a protocol that uses client-server publish-subscribe communication is fast, portable and efficient[25]. A publish-subscribe pattern-based system depends on a centralized server, defined as a message broker (server), to which access points (clients) are associated. From the client's view, the broker sends and receives all messages in this kind of model. At the point when MQTT customer distributes application information to an explicit topic, it basically sends PUBLISH packet to the broker. This packet includes real application information and topic yet, in addition, other significant data, for example, held a flag and QoS level.[26].

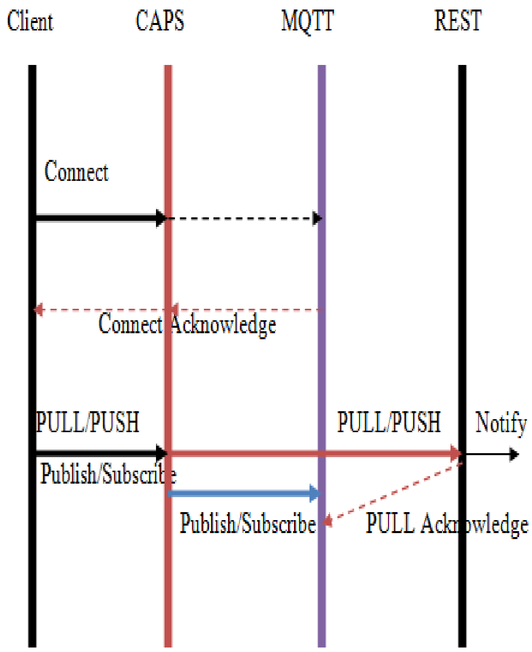

Fig. 7. Sequence diagram for Context-Aware Publish-Subscribe (CAPS)

While MQTT user subscribes to the subject, it sends it to the broker SUBSCRIBE packet. Such a packet simply includes a list of QoS-level topics that the user would like to subscribe to. A client may define the degree of QoS individually of topics and diverse clients.[27].

MQTT is the unbelieving protocol for data. It implies that the fundamental application information contained in the PUBLISH packet's content component included everything from binary representation to a high-level format like JSON (JavaScript Object Notation) or CSV (Comma-Separated Values), XML (Extensible Markup Language). The broker, who transfers messages among users, can convey the message without the ability to understand the real data in the content part of the request. Clients were responsible for processing the content [27].

The above figure represented the details about Context-Aware Publish and Subscribe (CAPS) approach. In this, we have mentioned about the publish, subscribe, connect, Push, pull methods. Here, if a client wants to get information from a server, we have connected then will get the acknowledge from the server, so here MQTT is used to obtain the data from sensors and REST is used to response the request from the client. Among this CAPS is used to allocate the request from the client, whether it is for push, pull or publish and subscribe so based the users request the caps approach can work accordingly. The sequence of the CAPS process is illustrated in Figure-6.

VI. IMPLEMENTATION

There may be numerous methods to extracting meaning from IoT data; around here we are studying various methods such as JSON, MQTT, MongoDB, RESTfull API.

The framework has been constructed so for to manage the present context, that is, it contains the information entities that describe the condition of the real-world objects at a given time. You see from this concept-context is only comfortable with the system present state. It's not the liability of any of the internal elements to comment on the system's historical state. The context depends on the last calculation that has been sent to the context broker by each sensor. To do this, the new architecture will need to be extended whenever the scope is updated to deal with changes in the database's status. Among each Smart Approach, though, the importance of each type of entity may vary and it may be appropriate to test entities and attributes at different rates.

Import and export JSON and CSV archive to implement MongoDB. By default, information is exported in JSON format here, but you can also export it to CSV (comma-separated value). Use the command mongo export to export MongoDB data. It enables you to export an extremely great-grained export to define a list, a set, a region, and also to export a request.

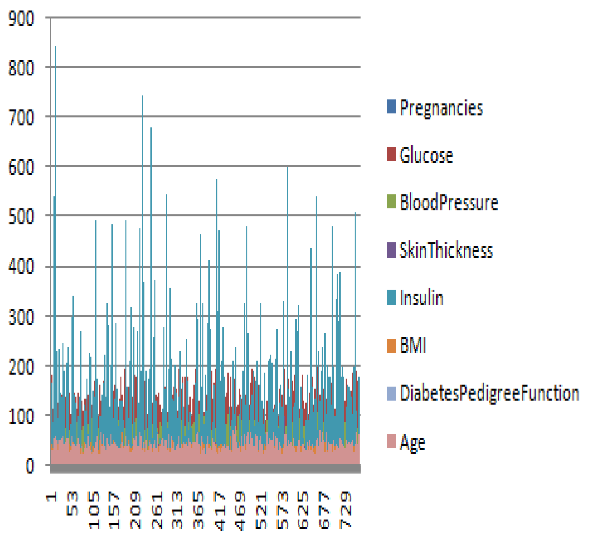

Fig 8. Attributes of GDM Patients with profile

Fig 8 indicates about the time is taken for data retrieval of different sources of sensors. It can execute various files with the various time taken as stated. The X-axis represents the no of rules that have taken and Y-axis represents time m/s. under the result in terms of data retrieval has varied compared to existing work.

Why API matter to IoT

- APIs are capable of gluing and combining product
- APIs enable users to create context applications
- APIs are really a collection of software application developing procedures, protocols, and tools.

Methods:

- **GET:** GET is used to request information from a source listed.
- **PUT**: PUT sends resource information to an updated database.
- **POST:** POST sends information to a database to build or update a resource.
- **DELETE:** DELETE eliminates the source server resources.

Importing the Data

Mongo database Import syntax

➢ mongoimport --host <host_name> --username <user_name> --password <password> --db<database_name> --collection <collection_name> --file <input>

Exporting the Data

Mongo database Export syntax

➢ mongoexport --host <host_name> --username <user_name> --password <password> --db<database_name> --collection <collection_name> --out <out>

What something requires building in an Internet of Things system? There is trick to move data efficiently and quickly — so API is the heart of any IoT implementation.

VII. CONCLUSION

This article proposes as a part of an ongoing IoT based context-aware platform. With the aid of rule-based modeling, we concentrate in this paper on context acquisition and building context profiles for contextualization processing. Context acquisition can translate the raw- data into high-level context information based on the CAPS approach. The proposed system offers sensing, processing and representing the data. We have created a context profile based on the context types and attributes. Contextualization has done through the CAPS approach. We have offered an evaluation function of the Gestation Diabetics Mellitus (GDM) in healthcare domain. Several prospective realistic applications of the proposed architecture among others are healthcare, car parking, and recommender system. In future work, we like to apply in requirements and constraints for different applications.

REFERENCES

1. B. Schilit and M. Theimer, “Disseminating active map information to mobile hosts,” *Network, IEEE*, vol. 8, no. 5, pp. 22 –32, sep/oct 1994. [Online]. Available: http://dx.doi.org/10.1109/65.313011.
2. Perera, C., Zaslavsky, A., Christen, P., & Georgakopoulos, D. (2013). Context-aware computing for the internet of things: A survey. *IEEE communications surveys & tutorials*, *16*(1), 414-454.
3. Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A., & Riboni, D. (2010). A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing*, *6*(2), 161-180.
4. A. Dey, Understanding and using context, Personal Ubiquitous Comput. 5 (1) (2001) 4–7
5. Giusto, D. (2010). A. lera, G. Morabito, l. Atzori (Eds.) The Internet of Things.
6. Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). A survey on sensor networks. *IEEE Communications magazine*, *40*(8), 102-114.
7. Ashton, K. (2009). That ‘internet of things’ thing. *RFID journal*, *22*(7), 97-114.

8. Vermesan, O., Friess, P., Guillemin, P., Gusmeroli, S., Sundmaeker, H., Bassi, A., ... & Doody, P. (2011). Internet of things strategic
9. research roadmap. *Internet of things-global technological and societal trends*, *1*(2011), 9-52.
10. Sundermann, C. V., Domingues, M. A., Conrado, M. da S., & Rezende, S. O. (2016). Privileged contextual information for context-aware recommender systems. *Expert Systems with Applications*, *57*, 139–158. https://doi.org/10.1016/j.eswa.2016.03.036.
11. Van der Veen, J.S.; Van der Waaij, B.; Meijer, R.J. Sensor data storage performance: SQL or NoSQL, physical or virtual. In Proceedings of the 2012 IEEE Fifth International Conference on Cloud Computing, Honolulu, HI, USA, 24–29 June 2012; pp. 431–438.
12. Gomez, C.; Oller, J.; Paradells, J. Overview and evaluation of Bluetooth low energy: An emerging low-power wireless technology. Sensors 2012, 12, 11734–11753.
13. Kale SD, Kulkarni SR, Lubree HG, Meenakumari K, Deshpande VU, Rege SS, et al. Characteristics of gestational diabetic mothers and their babies in an Indian diabetes clinic. J Assoc Physicians India. 2005;53:857–63.
14. . Haver MC, Locksmith GJ, Emmet E. Irregular menses: An independent risk factor for gestational diabetes mellitus. Am J Obstet Gynecol. 2003;188:1189–91. [PubMed] [Google Scholar]
15. 5. Berkowitz GS, Lapinski RH, Wein R, Lee D. Race/Ethnicity and Other Risk Factors for Gestational Diabetes. Am J Epidemiol. 1992;135:965–73.
16. Keshavarz M, Cheung NW, Babaee GR, Moghadam HK, Ajami ME, Shariati M. Gestational diabetes in Iran: Incidence, risk factors and pregnancy outcomes. Diabetes Res Clin Pract. 2005;69:279–86.
17. Shaw JE, Sicree RA, Zimmet PZ 2010 Global estimates of the prevalence of diabetes for 2010 and 2030. Diabetes Res Clin Pract 87:4 –14.
18. Bevier WC, Fischer R, Jovanovic L 1999 Treatment of women with an abnormal glucose challenge test (but a normal oral glucose tolerance test) decreases the prevalence of macrosomia. Am J Perinatol 16:269 –275
19. Acciaroli, G.; Vettoretti, M.; Facchinetti, A.; Sparacino, G. Calibration of minimally invasive continuous glucose monitoring sensors: State-of-the-art and current perspectives. Biosensors 2018, 13, 24.
20. Lane, N. D., Miluzzo, E., Lu, H., Peebles, D., Choudhury, T., & Campbell, A. T. (2010). A survey of mobile phone sensing. *IEEE Communications magazine*, *48*(9), 140-150.
21. Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosof, B., & Dean, M. (2004). SWRL: A semantic web rule language combining OWL and RuleML. *W3C Member submission*, *21*(79), 1-31.
22. Román, M., Hess, C., Cerqueira, R., Ranganathan, A., Campbell, R. H., & Nahrstedt, K. (2002). A middleware infrastructure for active spaces. *IEEE pervasive computing*, *1*(4), 74-83.
23. Ko, K. E., & Sim, K. B. (2008, October). Development of context aware system based on Bayesian network driven context reasoning method and ontology context modeling. In *2008 International Conference on Control, Automation and Systems* (pp. 2309-2313). IEEE.
24. McBride, B. (2004). The resource description framework (RDF) and its vocabulary description language RDFS. In *Handbook on ontologies* (pp. 51-65). Springer, Berlin, Heidelberg.
25. Heflin, J. (2007). An introduction to the OWL web ontology language. *Lehigh University. National Science Foundation (NSF)*.
26. mqtt-v3.1.1-plus-errata01, Organization for the Advancement of Structured Information Standards (OASIS), 2015, 81 p. Available (referred 13.1.2017): http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.pdf.
27. Publish & Subscribe, MQTT Essentials, dc-square GmbH, Available (referred 13.1.2017): http://www.hivemq.com/blog/mqtt-essentials-part2-publish-subscribe.
28. MQTT Publish, Subscribe & Unsubscribe, MQTT Essentials, dc-square GmbH, Available (referred 13.1.2017): http://www.hivemq.com/blog/mqtt-essentials-part-4-mqtt-publish-subscribe-unsubscribe

AUTHORS PROFILE

Deeba K received B.Sc.,(computer science) 2009, M.sc.,(Computer science) 2011 , M.Phil.,(Computer science) 2012. She is doing her Ph.D in computer science and engineering at Vellore Institute of Technology, Vellore,Tamilnadu, India. Her research interest includes context aware system, Middleware, Internet of things and Machine learning techniques (deebakanmani@gmail.com).

Dr. RA K Saravanaguru has been associated with Vellore Institute of Technology (VIT), Vellore since June 2004 and presently working as Associate Professor in School of Computer Science and Engineering (SCOPE) and Assistant Dean Academics. He has sixteen years of teaching experience. He completed his Ph.D in Computer Science and Engineering in the field of Context aware middleware for vehicular adhoc network. His area of interest includes context aware systems, middleware, web services, VANET, data science and network security.