Context-Aware System for Indoor Localization

M. Robiul Hoque[1,*], Shomako Bosu[1], M. Humayun Kabir[2], and Md. Aktaruzzaman[1]

[1]Department of Computer Science and Engineering, Islamic University, Kushtia-7003, Bangladesh.

[2]Department of Electrical and Electronic Engineering, Islamic University, Kushtia-7003, Bangladesh.

robiul@cse.iu.ac.bd, shomabosu@gmail.com, humayun@eee.iu.ac.bd, aktaruzzaman@iu.ac.bd

Abstract

User location is an important information to provide personalized service in any domain of interest. In this paper, we present a context-aware system to find location of a user at office domain. Hence, smartphone of the user transmits Wi-Fi signal and our system identifies the user by matching MAC address of the smartphone. Wi-Fi signal strength is used to calculate the distance of user from office room. Additionally, PIR sensor value and office time are used to increase localization accuracy as well as minimize the conflict of user location. Raspberry pi 3, a low-cost embedded platform, is used to collect and process sensed information, generate low-level context, and reason user location from available contexts. Demonstration result of our system shows an excellent performance within our domain of interest.

Keyword: Context-Awareness, Indoor-Localization, Sensor, Service, Rule-Base, Smartphone.

1. Introduction:

In recent year, indoor localization technique is becoming increasingly important for a large number of applications, such as healthcare, homecare, monitoring, tracking, smart space, etc.[1]. Context aware is the ability of a system or system component to gather information about its environment at any given time and adapt behaviors accordingly [2]. Where, context is any information which is relevant to a given entity, such as a person, a device or an application. Context aware system is mainly used to solve problems in the field of ubiquitous computing [3, 4]. However, localization of a moving person in indoor space is a vital part of a ubiquitous computing system. So, context aware system can be employed to detect location in indoor environment for better accuracy and efficiency. At the present time, outdoor positioning technology has matured enough. Global Positioning System (GPS) has been widely used in outdoor environments and is well positioned. But as GPS technology mainly reside on signal propagation in air, it is drastically affected by buildings and their complex infrastructures and thus are not suitable for indoor localization. For indoor localization, most of cases, user needs to bring an identification RFID tag which is making extra hazard for the user. Beside RFID, various wireless communication systems like Bluetooth, Zigbee and Wi-Fi signal are used. Wi-Fi access points can be seen in almost every indoor system and almost all smartphone devices have a built-in Wi-Fi transceiver module. As a result, Wi-Fi indoor localization has become an interesting research topic in the modern time.

Now a days, almost every people uses smartphone for personal communication. So, it is better alternative to use smartphone as Wi-Fi hosts and an identification module instead of the extra RFID tag. The main aim of this research is to find an exact location in indoor environment of a user using wireless network. Because, location is an important context of other context aware systems to provide personalized service [5]. In this paper, our system detects surrounding Wi-Fi device and provide their RSSI (Received Signal Strength Indication), Basic Service Set Identifier- BSSID (MAC addresses). This information is used to generate contexts, such as, user distance from the system unit and the user identity. This system also uses movement information from PIR sensor and current system-time as context. After getting all these contexts (like: time, movement, identity, distance etc.), the system uses rule-based reasoning [6] to get high-level context which will be the exact location of a user.

Related Works

Although many researchers have presented localization system based on different technologies, the development of localization in the indoor environment is still a complex task in the sense of accuracy and precision, and seamless user involvement. Wireless Sensor Network (WSN) is well known technology to implement indoor localization system [7], where Radio Frequency Identification (RFID) is used most frequently for object tracing [8-10]. Analysis of previous works shows that no single solution can meet all criteria for widespread implementation of indoor localization system [11]. Mainly Accuracy and power consumption are the big challenges for WSN based systems. On the other hand, RFID based system needs several RFID readers spread out in the environment in known positions which increase the cost and complexity for implementation of the system.

Vision based system uses camera in several fixed positions or with a moving object for localization. This method mainly concentrates the target to localize an object without recognition [12, 13]. The cost of this technique is still high but new technologies are gradually helping to reduce this constrain. Bluetooth, Ultrasound, Infrared and Zigbee signals are also used for wireless localization in different researches that show diverse level of performances [1]. Here, we proposed a context aware system for localization in room level and implemented using low cost embedded platform (Raspberry pi 3) and Python as programming language. We also incorporate some other well-known technologies such as, context representation using FOPL, ontology based context modeling, and rule-based reasoning. As a result, this system shows satisfactory accuracy for localization with more flexible and scalable abilities. So, it will be possible to implement this system in different domains easily.

2. Proposed system architecture:

The system architecture is a collection of software modules organized in hierarchy fashion to perform particular task [14]. Our proposed system architecture consists of several functional components as shown in Fig. 1. Following subsections represent each module of this architecture in briefly.

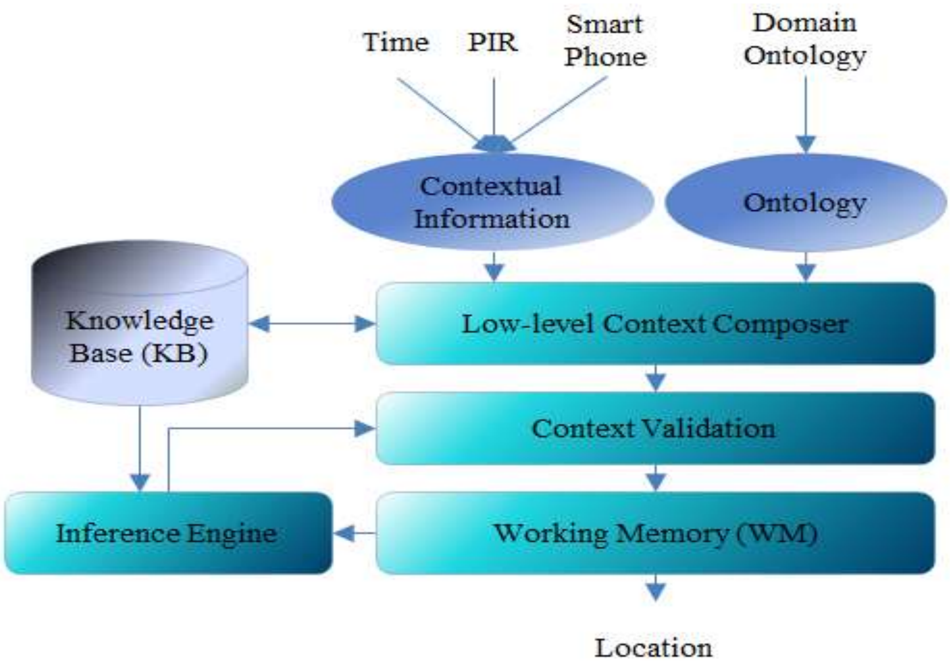
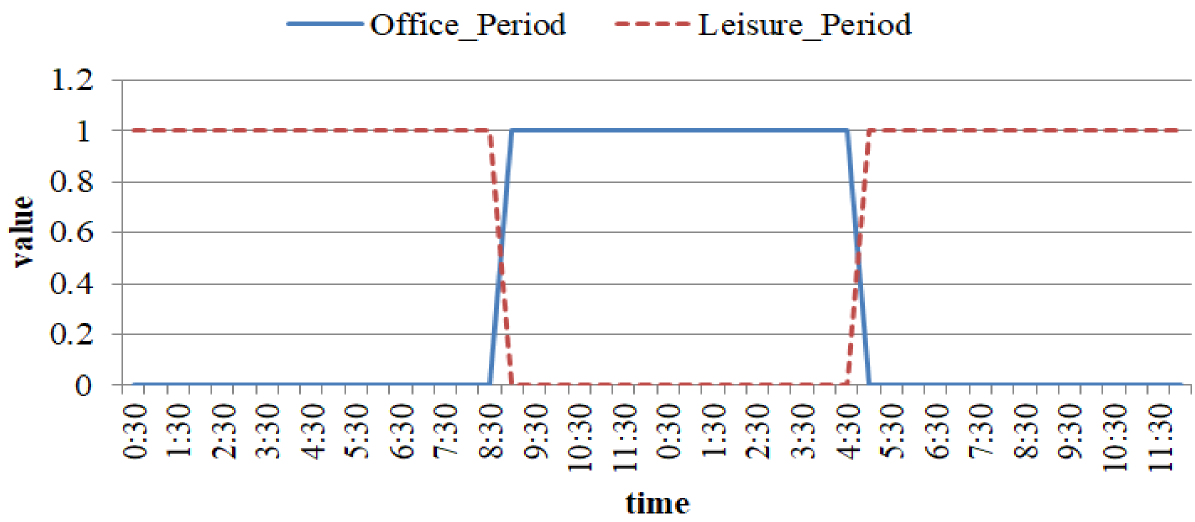


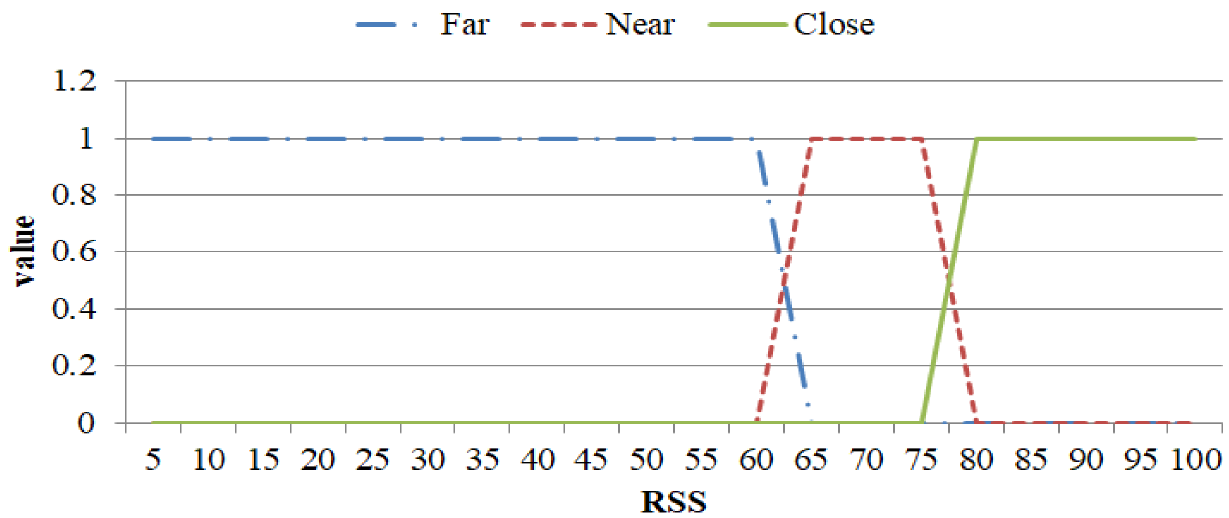
Fig. 1. Context-aware localization system architecture

2.1. Contextual Information

Proposed system collects contextual information through motion sensor (PIR), Wi-Fi base received signal strength with BSSID (MAC Address), system time, and these information is mapped using Fuzzy function to represent in literally. For example, system time is expressed in numeric format (e.g., 11:30 AM), but we need to know this time is office time or not. So, we divided a day by two different time periods in our office scenario. Such as, *office_period* and *leisure_period*. Similarly, location is considered as *room425* and *outside* in linguistic. Instances of different context in detail will be described in Ontology subsection. *Fuzzy function:* Fuzzy function provides mapping between the numeric data and the instance of context entities, such as mapping of the Wi-Fi signal strength to distance in linguistically. Because, people commonly use linguistic expression rather than numeric value to express their feel concerned about the real world. For example, to express their understanding of distance, people use phrases such as close, near, far, and so on, instead of referring to the distance in meter. To fuzzify the sensing data, we defined fuzzy sets and membership functions as shown in Fig. 2.



(a) Time function



(b) Distance function

Fig. 2: Membership functions for (a) time and (b) distance.

2.2. Ontology:

Ontology [15] represents semantics, concepts and relationships among the context entities. It provides domain specific knowledge to context-aware system [16].

In our domain, the entities are mainly categorized into user, time, location, distance and motion classes. Each class can be divided into several subclasses. This model defines the class hierarchy, object properties, data type properties. As our office domain is not too large, we primarily implemented this context model using relational database. A part of this ontology is shown in fig. 3.

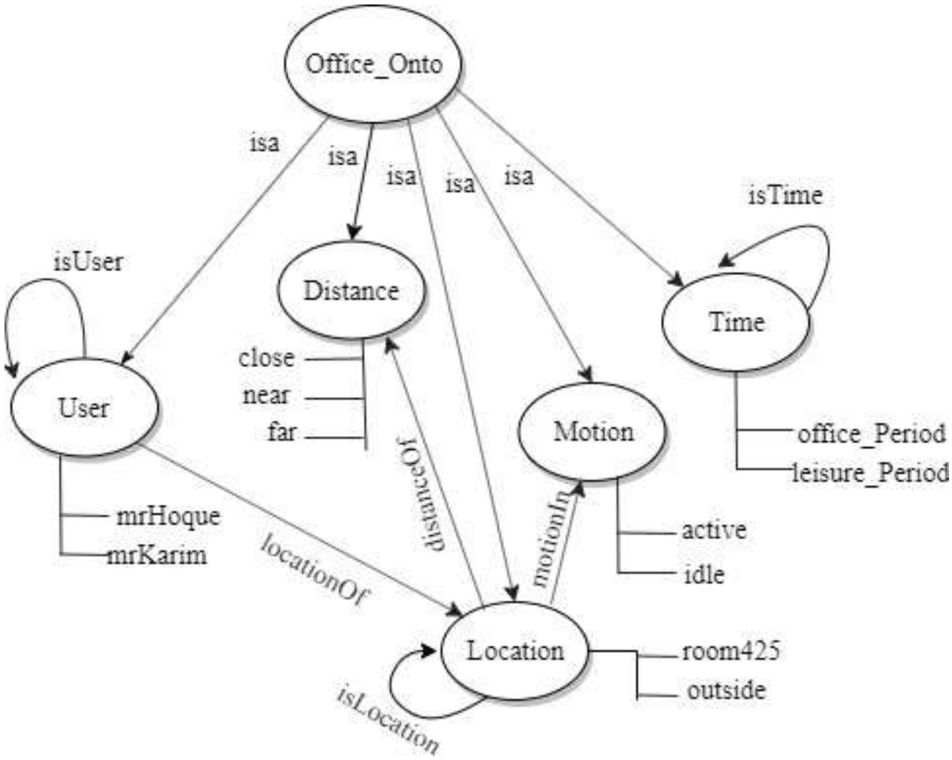


Fig. 3: A part of office ontology.

2.3. Low-level Context Composer

A low-level context is generated from sensing and/or defined data. Low-level context composer generates low-level context using contextual information and ontology. Firstly, this module gets symbolic context instance from contextual information. Secondly, it finds relationship between concepts of corresponding instances in ontology. Finally, composes low-level context by combining instances and relationship as like, $r(i_1, i_2)$, where r is the relation between concept of the instances i_1 and i_2. In case of single instance, context format will be $r(i)$. Here, context is represented in formal way as like as first-order predicate logic, because it influences to produce formal context as the result of reasoning [17]. For example, let *active* is an instance of motion concept and *room425* is an instance of location concept of the office ontology, and the relation between them is *motionIn*. So the generated low-level context is *motionIn(room425, active)*.

2.4. High-level Context Generation

High-level context is generated by reasoning process from available contexts. Reasoning engine is invoked at every time, when a new context is added in WM. This process will be repeated until the system is terminated. Reasoning engine computes high-level context according to the rules. In this system, user location is inferred as high-level context. An example of high-level context is *locationOf(mrHoque, room425)*. If none of the given rule is satisfied then it generate high-level context indicating location of the user is *outside* like, *locationOf(mrHoque, outside)*. The reasoning process is summarized in Algorithm 1.

Algorithm 1: Generation of high-level context

```
1. List up all variables of a rule
2. Find all instances of each variable from WM
3. Recursively put all instances in the rule and check whether the rule is
   satisfied or not
   a. If the rule is satisfied
      Generate a high-level context accordingly
      End if
4. Repeat step 1 to 3 for each rule
5. If no rule is satisfied, generate location of the person is outside.
   End if.
          6. Send the high-level context to WM through validity check
7. Return
```

Algorithm Complexity: Algorithm 1 is used to find user location by matching each rule with available contexts in WM. If no. of variable in a rule is m, and no. of instance of a variable is n (where n is a finite integer) then the time complexity of this algorithm in general can be express as below.

For worst case: $O(mn)$

For best case: $\Omega(m)$

2.5. Working Memory

Contexts, in this architecture, are stored in working memory (WM) through validity check. Where, WM is a temporary storage to hold context to perform context inference. If a location context is updated as a result of inference, then this location context is provided as output from WM.

Context validation

This module checks validity of context and maintains context consistency in WM by discarding redundant or duplicate context.

2.6. Inference Engine

Inference engine is used to reason new high-level context from available contexts in WM. To develop this engine, rule-based inferring technique is used. During reasoning process, rule's variables are replaced by context instances from WM.

Possible all combination for each rule is checked and if 'if part' of a rule is matched then the 'then part' will be fired and a high-level context is produced accordingly. Reasoning engine infers user's current location as high-level context. High-level context is also added in to WM through validity check as like as low-level context.

Rules: In the reasoning process, rules represent derivation axioms that are used by the inference engine to derive high-level context. Several contexts with variable are combined by ^ (AND) to form a rule. The rules are predefined that means they are defined by the user and stored before booting up the system. Some examples of user defined rules [18] are shown in Table 1.

Table 1: Partial user defined rules

SN	Rule
1	if isUser(?p)^isLocation(?l) ^distanceOf(?l,closer) ^motionIn(?l,active) ^isTime(office_period) then locationOf(?p,?l)
2	if isUser(?p) ^isLocation(?l) ^distanceOf(?l,near) ^motionIn(?l,active) ^isTime(office_period) then locationOf(?p,?l)
3	if isUser(?p) ^isLocation(?l) ^distanceOf(?l,closer) ^motionIn(?l,idle) ^isTime(office_period) then locationOf(?p,?l)

3. Implementation

Context-aware indoor localization system produces location context of user with the help of a number of hardware and software module as mention below-

Hardware: Raspberry pi 3, PIR sensor, Smartphone are used as hardware modules for this research. All modules are briefly described in below subsections. *Raspberry pi 3*- Raspberry pi 3 is simply a computer on a single board [19]. For the pi to start working we need to install operating system first. The pi has dedicated OS for it. We used Ubuntu-mate as OS for this research.

PIR sensor- PIR sensor detects a human being moving around within approximately 10m from the sensor. PIR are fundamentally made of a pyroelectric sensor, which can detect levels of infrared radiation [20]. *Smartphone*- This system analyzes Wi-Fi signal strength from authorized smartphone, which is carrying by the user, to estimate user-distance from office room. Beside this, Media Access Control (MAC) address of the smartphone is used as the identifying information of the user. Where, MAC is a unique hardware identifier number that is assigned on each device by the manufacturer. MAC addresses are 48 bits in length, divided in 24 bits vendor code and 24 bits serial address and the format of MAC address is MM:MM:MM:SS:SS:SS [21]. MAC address is very important to identify each device in a network and to make local Ethernet network functional.

Software: To implement this system, we used Ubuntu Mate as the operating system for raspberry pi 3. Python is used as the programming language of this system. Beside these, necessary database is developed using SQLite.

Ubuntu mate- Ubuntu mate [22] is an operating system which is a Linux distribution for raspberry pi.

Python- Python is a general purpose, dynamic, high level and interpreted programming language [23]. It supports Object Oriented programming approach to develop applications.

SQLite- SQLite is an in-process library that implements a self-contained, server less, zero configurations, transactional SQL database engine [24]. SQLite supports most of the query language features found in SQL92 SQL2) standard. SQLite is available on UNIX (Linux, Mac OS-X, Android, iOS) and Windows (Win32, WinCE, WinRT). *Domain*: To test our system, we consider a personal office room as test environment. The target of this system is to determine whether a personnel is present at his/her office room or not.

Testing setup: The overall hardware setup is shown in Fig. 4. For demonstration, we request to a volunteer to act as a personnel who come in the room and go outside of the room several times. For several position of the volunteer, system put on color LED to indicate the location of the volunteer. For example, red indicates outside, whereas green color indicates he is inside the room. Beside this, we also recorded volunteer's location manually on paper during testing with time for comparing system output. As well as, we also store current location context along with other available contexts for output validation and further research.

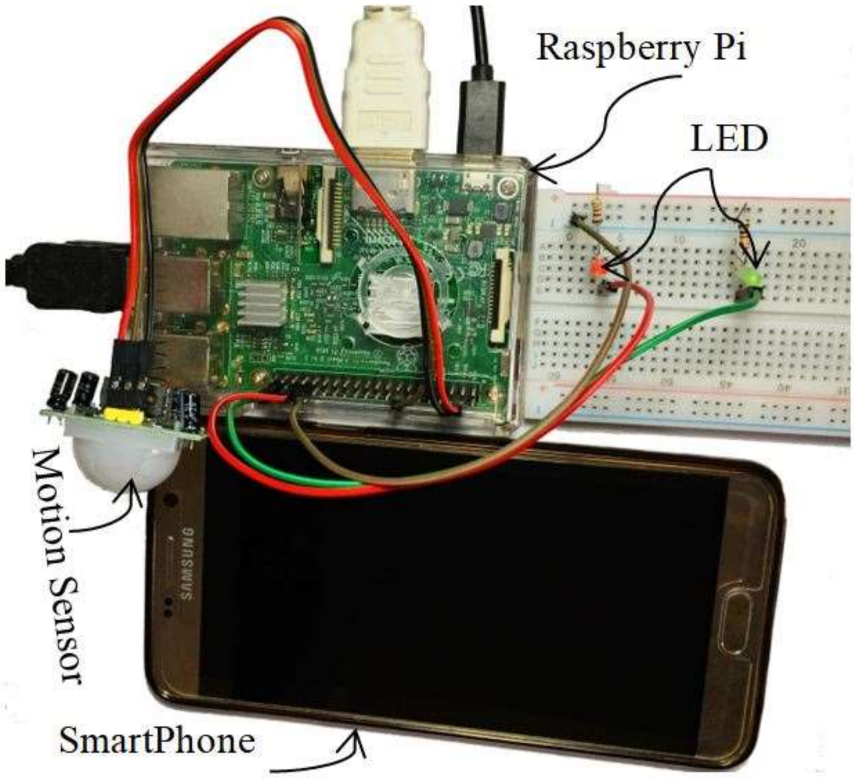


Fig. 4: Test environment setup.

4. Result and Discussion

In this research, we have incorporated context-aware paradigm for localization. We tested this system in personal office room. It is implemented using Python language on Raspberry pi. This system senses data frequently with a short period of predefined delay between each sense. Available contexts are stored in database when at least one of the context values is changed in WM. For validating the system, we compared manual result with proposed system's output. And it shows that, system produces almost 100% correct result for our domain of interest. Sometime, localization accuracy can reduce for erroneous sensing information. For example, variation of Wi-Fi signal strength, user is staying inside without any movement which can mislead localization accuracy. Beside this, system also can produce slow response compare with user faster movement. Therefore, for better result, steady and high speedy Wi-Fi network is always appreciated.

5. Conclusion:

This paper has illustrated context-aware localization framework with its functionalities in an office room environment. This system is implemented on raspberry pi platform with python programming language. It provides relax to user from carrying any other identifying tag except user's smartphone. This system senses information from user smartphone and PIR sensor and system time to generate contexts using FOPL representation. Then it computes user location based on these contexts using predefined rule. Demonstration results of this system show excellent performance with room level accuracy. The most significant benefit of this system is it is easy expandable that means it can be implemented in other domain easily. It needs to change only domain ontology and inferring rules according to the new domain. Therefore, this study helps to develop different location based applications in different domains such as Healthcare, Smart-home, infant and elder care, tourism and so on.

6. Future Works:

Although the performance of this system is quite good for personal office with single room, we have to resolve some relevant challenges to make this system more reliable and robust. In future, we will evaluate the performance of the system in different domains such as office, hostel or class room considering multiple rooms with multiple users.

After determining the location, system can provide the proper services to the right user at the right time, in the right place, and on the right device based on the available contexts. So, we will design and implement different location based services to facilitate daily life activities of user. For example, system will adjust the room temperature or light intensity according to user presence and preference. We will also implement machine learning technique on stored data for better accuracy.

Conflicts of Interest: The authors declare that there is no conflict of Interests regarding the publication of this paper.

7. References:

1. Luca Mainetti, Luigi Patrono and Ilaria Sergi, "*A Survey on Indoor Positioning Systems*", 22nd International Conference on Software, Telecommunications and Computer Networks (SoftCOM), Split, Croatia, 2014

2. Anind K. Dey and Gregory D. Abowd, " *Towards a Better Understanding of Context and Context-Awareness*", Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing, 1999, pp.304-307.

3. Mario Pichler, Ulrich Bodenhofer and Wieland Schwinger, "*Context-Awareness and Artificial Intelligence*", ÖGAI Journal, vol. 23, pp. 4-11, 2004.

4. M. Humayun Kabir, M. Robiul Hoque, Hyungyu Seo, and Sung-Hyun Yang, " *Machine Learning Based Adaptive Context-aware System for smart Home Environment*", International Journal of smart Home, Vo. 9, No. 11, 2015, P. 55-62.

5. Hyun-Wook Kim, M. Robiul Hoque, Hyungyu Seo, and Sung-Hyun Yang, "*Development of Middleware Architecture to Realize Context-Aware Service in Smart Home Environment*", Computer Science and Information Systems, Vol. 13 (2), pp. 427-452, June, 2016. https://doi.org/10.2298/CSIS150701010H.

6. M. Robiul Hoque, M. Humayun Kabir, Hyungyu Seo, and Sung-Hyun Yang, "*PARE: Profile-Applied Reasoning Engine for Context-Aware System*", International Journal of Distributed Sensor Networks, SAGE Publishing, Vol. 12(7), pp. 1-11, July, 2016. https://doi.org/10.1177/155014775389091.

7. L. Chen, E. Wu, and G. Chen, "*Intelligent Fusion of Wi-Fi and Inertial Sensor-Based Positioning Systems for Indoor Pedestrian Navigation,*" IEEE Sensors J., vol PP, issue 99, June 2014. https://doi.org/10.1109/JSEN.2014.2330573.

8. M. Maffia, L. Mainetti, and L. Patrono, "*Evaluation of potential effects of RFID-based item-level tracing systems on the integrity of biological pharmaceutical products*," Int. J. of RF Technologies: Research and Applications, vol. 3, no. 2, pp. 101-118, 2012. https://doi.org/10.3233/RFT-2012-022.

9. G. Calcagnini, F. Censi, M. Maffia, L. Mainetti, E. Mattei, L. Patrono, and E. Urso, "Evaluation of thermal and nonthermal effects of UHF RFID exposure on biological drugs," IEEE Transactions on Information Technology in Biomedicine, vol. 16, no. 2, pp. 1051-1057, 2012. https://doi.org/10.1109/TITB.2012.2204895

10. L. Catarinucci, R. Colella, M. De Blasi, V. Mighali, L. Patrono, and L. Tarricone, "*High performance RFID tags for item-level tracing systems*," Proc. of International Conference on Software, Telecommunications and Computer Networks (SoftCOM 2010), Split, Dubrovnik, pp.21-26, 23-25 Sept. 2010.

11. N. Li and B. Becerik-Gerber, "*Performance-based evaluation of RFID based indoor location sensing solutions for the built environment*," Adv. Eng. Inf., vol. 25, pp. 535–546, Aug. 2011. https://doi.org/10.1016/j.aei.2011.02.004

12. T.-H. Tsai, C.-H. Chang and S.-W. Chen, "*Vision based indoor positioning for intelligent building,* 2nd International Conference on Intelligent Green Building and Smart Grid (IGBSG), Prague, Czech Republic, 27-29 June 2016. https://doi.org/10.1109/IGBSG.2016.7539419

13. W. Elloumi et al., "*Indoor navigation assistance with a Smartphone camera based on vanishing points*," Int. Conf. Indoor Positioning and Indoor Navigation (IPIN), Montbeliard, FR, pp. 1-9, 28-31 Oct. 2013.

14. M. Robiul Hoque and M. Humayun Kabir, "*Context Management Framework for Context-Aware Service*," Journal of Applied Science and Technology, Vol-11, No.-01, PP.30-36, February 2019, Islamic University, Kushtia, Bangladesh.

15. D. Fensel, "*Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*," Springer, 2003.

16. T. Gu, H. K. Pung, D. Q. Zhang, "*A Service-Oriented Middleware for Building Context-Aware Services*," Journal of Network and Computer Applications, Vol. 28(1), 2005, pp.1-18. https://doi.org/10.1016/j.jnca.2004.06.002

17. E.-S. Cho, T.-S. Yoon, J.-H. Choi, J.-Y. Paik, S. Helal, "*An Integrated Formal Model for Context-Aware Systems*," Proceedings of the IEEE 37th Annual Computer Software and Applications Conference Workshops, 2013, pp.163-168.

18. J.-H. Jang, H.-G. Seo, S.-H. Yang, "*A Rule-Making Method Using State Diagram for Rule-Based Context-Aware System in Smart Home Environment*," Proceedings of the International Symposium on Advanced and Applied Convergence, 2014, pp.138-141.

19. Raspberry pi model 3, "https://www.raspberrypi.org/products/raspberry-pi-3-model-b/"

20. PIR, "https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor/overview"

21. MAC address, "https://www.lifewire.com/introduction-to-mac-addresses-817937"

22. Ubuntu Mate, "https://ubuntu-mate.org/raspberry-pi/"

23. 2Python, "https://www.python.org/"

24. Sqlite, https://www.sqlite.org/index.html