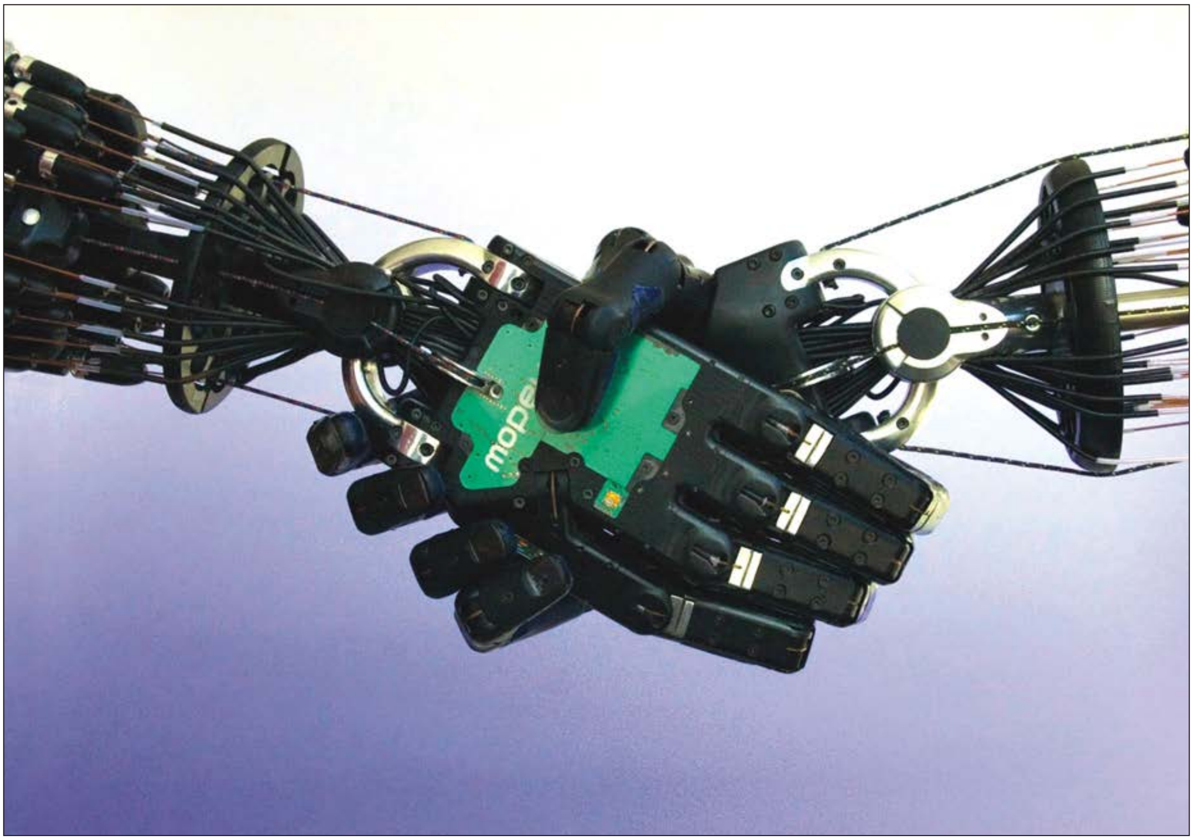

Journal of Artificial Intelligence

ISSN 1994-5450

Journal of Artificial Intelligence 7 (4): 145-160, 2014
ISSN 1994-5450 / DOI: 10.3923/jai.2014.145.160

Smarter Environments: A Compact Service-Oriented Framework for Context-Awareness

[1]Veeramuthu Venkatesh, [2]Pethuru Raj, [1]V. Vaithiyanathan and [3]Rengarajan Amirtharajan
[1]School of Computing, SASTRA University, Thanjavur, 613401, India
[2]Global Cloud Center of Excellence, IBM India, Bangalore, 560045, India
[3]School of Electrical and Electronics Engineering, SASTRA University, Thanjavur, 613401, India

Corresponding Author: Rengarajan Amirtharajan, School of Electrical and Electronics Engineering, SASTRA University, Thanjavur, 613401, India Tel: +91 4362 264101 Fax: +91 4362 264120

ABSTRACT

Context-awareness is an essential feature for establishing and sustaining a variety of specific as well as generic smarter environments. However, the acts of gleaning context information from different and distributed sources and interpreting it for real-time and actionable insights are not easy tasks. Having understood this requirement, we have conceptualized a new-generation context-awareness framework in order to simplify and streamlining context-awareness, which is the distinct need for next-generation smarter spaces. In this study, we have laid down all the right and relevant details of the approach and the framework solution through a host of renowned use cases. Further on, we have vividly illustrated the performance evaluations for the authenticity and strategic success of our approach and the solution incorporated.

Key words: Smart environments, context-awareness, automata, service oriented architecture, security

INTRODUCTION

The indisputable and impending trend is that commonly found articles in our midst are being empowered to find, connect and interact with one another to spontaneously leverage their specific capabilities and competencies. That is, any ordinary artifacts in our personal as well as professional environments seamlessly join in the mainstream computing to set the ball rolling to develop and deliver cognition-ignited, context-aware and people-centric applications and services. This game-changing empowerment is seeing a nice and neat reality with the adaption and adoption of a growing array of pioneering technologies (extreme connectivity, service-enablement, integration, orchestration, policy-based maneuvering, scores of edge technologies, ambient communication, sentient computing, ubiquitous sensing, perception and vision, knowledge engineering, information visualization, etc.).

Businesses have seen enough technologies-inspired automation and acceleration in the past decades. However, the future of those promising and potential technologies benevolently lies with all the tangible domains that are directly and indirectly aligned with human lives. We often read, hear and sometimes feel scores of smart (Albreshne *et al*., 2013; Islam *et al*., 2013b; Hong *et al*., 2013; Lee *et al*., 2014) artifacts (digitalized objects/sentiment materials), smarter devices, services and applications, etc. Increasingly, not only IT software and hardware solutions but also every kind of electronics in our everyday environments are being systematically empowered to be deeply and extremely connected, collaborative and cognitive. There are promising and potential edge technologies emanating and evolving fast in order to fulfill these next-generation requirements. Another noteworthy trend is that the digitalization idea has been penetrating and percolating into every single industry domain these days and hence all kinds of common, casual and cheap items in our everyday environments are being instantaneously turned and tuned into digitalized entities. The classic disciplines of networking and service-enablement are gaining a lot of ground thereby, newer and nimbler things and tasks emerge for enhanced care, choice, comfort and convenience for people.

These technology-driven empowerments ultimately help every artifact in our places to find, bind and cooperate with one another in the vicinity and with remote articles (digitalized objects, electronics devices, household items, medical instruments, manufacturing machines, kitchen ware and utensils, software applications and data in cloud platforms and infrastructures) via networks. The result is the constant eruption of hitherto unforeseen services that are people-centric, situation-aware, knowledge-packed, service-driven and sophisticated. There are a variety of platforms, middleware (Lopes *et al*., 2014; Warriach, 2013; Lee *et al*., 2012), databases, tools, engines and containers in order to build, deploy and deliver next-generation connected, cognizant and cognitive applications. Ultimately, people will come across scores of smart, real-world and real-time applications in an affordable, accelerated, adaptive and risk-free fashion. Newer technologies facilitating all these noteworthy advancements are for enabling everything is to be self, surroundings and situation-aware. Ultimately, our daily spaces are turned and tuned to be smarter in their offerings, operations and outputs. In this study, we have described a fresh context-aware approach and an implementation framework for the timely extraction of context information in real-time to plan and proceed with appropriate actions in any environments (homes, hospitals, hotels, class rooms, offices, manufacturing floors, transports, etc.).

CONTEXT-AWARE COMPUTING

IT organizations and academic institutions have been collaboratively working together and bringing forth newer ad nimbler computing paradigms in order to accomplish widely changing expectations of businesses and users. However, thus far, the reality is that computing systems have been preprogramed to do certain things perfectly. That is, software applications on receiving specific inputs are capable of producing predetermined outputs (Ntanos *et al*., 2014; Zhang *et al*., 2013). But this sickening position is bound to change dramatically in the days to unfold. That is, futuristic compute machines can take several types of internal as well as external input data from distributed and disparate sources to produce smarter results that are more aligned to humans' needs (timely and spatial). That is, users' context/situation is being flawlessly captured and profusely leveraged by compute clusters to produce right and relevant results for humans.

User context is being primarily decided based on the implicit capability of establishing seamless and spontaneous connectivity and integration with a dynamic pool of software services and applications (social, enterprise, mobile, embedded, web and cloud), multiple data sources, scores of devices, appliances and sensors in the users' environment. In a nutshell, the fast-emerging paradigm of context-aware computing is to produce context-sensitive outputs for human beings to act upon that with all the clarity and confidence. Figure 1 present a bevy of decision-enabling contextual information vividly and illustrates how different aspects of one's situation is being quickly captured and utilized for arriving at actionable insights.

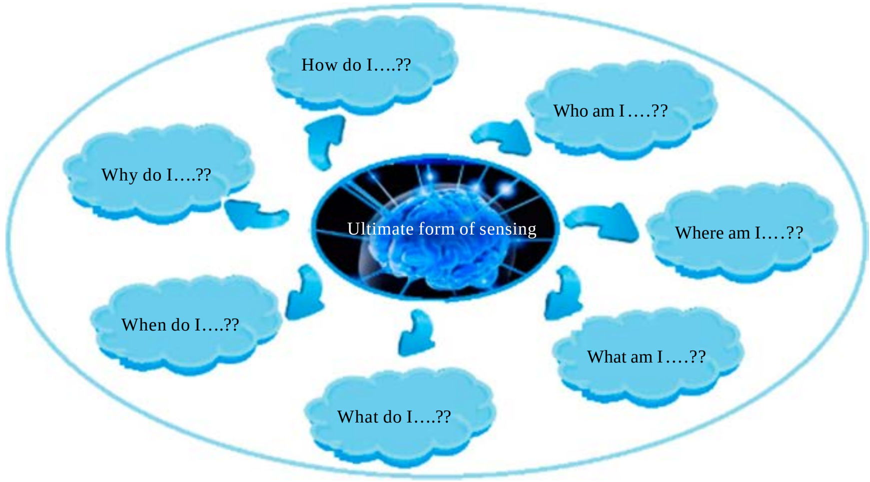

Fig. 1: A bevy of decision-enabling contextual information

EMERGENCE OF SMARTER ENVIRONMENTS

Increasingly, there are a growing array of smart sensors, controllers and actuators being deployed in any personal as well as professional environments. These digital entities and differently abled electronics are capable of extracting highly varying measurements of environment parameters in time and transmitting them to central systems by forming *ad hoc* networks. Increasingly, machines in our daily environments are getting instrumented and interconnected to be intelligent in their actions and reactions. Further on, cloud-based services and applications are being used to empower these entities to be distinct in their behavior. Thus, event-emitting sensors and devices at the ground level are integrated with remote, on-demand, online, off-premise and hosted cloud assets and applications via a host of service-oriented and standards-compliant middleware solutions.

A SAMPLE USE CASE

Let us consider the context of a home wherein each and every activity of an elderly people is to be monitored automatically. It becomes overhead to keep track of every movement of elderly people by appointing a care taker in a home. Instead, we can provide automated (Liang *et al*., 2014; Hatzi *et al*., 2012) environment with the help of sensors that makes the job easier. Suppose, to see if elderly person is present in his room or not, it is enough to monitor presence sensor that is attached to his body. Accordingly, we can either switch on or off the air conditioner. Also, the elderly people blood pressure levels, oxygen levels and so on in a particular moment that are recorded in sensors are continuously monitored remotely and if there is any abnormality, then it will be inform to a particular doctor to take care of that particular elderly people. This automation at the simplest is termed as “context aware computing” (Zhang *et al*., 2013).

Prior to the framework, we focus on the proven and potential benefits of finite state automata in formally representing and modelling the various interactions among various home-bound entities. It is possible to automate the tool-enabled direct transition from the automata representation into an executable code in any preferred programming language. There are specific languages for persisting finite automata representations. As there are more number of interconnected objects in smarter environments, the combination and permutation of their purpose-specific and user-defined interactions is bound to rise significantly. That is, the collaboration complexity, which is consistently on the climb, can be moderated considerably through the appropriate usage of the automaton concepts.

The literature survey focuses on the extensive range of practices, schematics, exemplars, schemes for middleware solvent connected to context awareness and also for the Internet of Things (IoT). This study is not only to take part, equate and collect the previous study done by earlier people who are all involved in this area and take it as an account to prize their findings and use their findings in internet of things, context awareness. Abowd *et al*. (1999) developed model and methods in this area as a beginning. After that, Chen and Kotz (2000) surveyed about in which application context aware can be used and how contextual can be rendered. Strang and Linnhoff-Popien (2004) analyzed the most important techniques in this domain. Molla and Ahamed (2006) collected information about middleware and its importance in the field of sensor networks. Bettini *et al*. (2010) have done a survey about context posturing and reasoning out some techniques instead of research projects. Bandyopadhyay *et al*. (2011) have done some survey on IoT and some middleware solutions. Makris *et al*. (2013) conducted a survey on context mobile networking. There are other researchers working on the various aspects of context-aware computing, which is the core and central concept for establishing and sustaining next-generation smarter environments that could understand and provide peoples' needs in time in an unobtrusive manner.

MATERIALS AND METHODS

Context-related concepts: In the above discussed use case (hospital automation), we just need to continuously track the data from the sensors in order to ensure that every patient receives the right treatment at the right time in a safe environment. For that to happen, we must consider all the context attributes, such as location, temperature and contexts related to some target. First, we need to identify the situations in the environment, like ward situation, patient situation, his health situation and emergency situation, etc. Then, we will identify related CAs (Context Attribute) in these situations and model these situations with CSP (Context Space Theory). To interrelate and to incorporate all the context aware concepts, the following descriptions have to be made:

- **Context Lacuna (C_{Lc}):** Context Lacuna is set of scopes of a specific domain that provides value range of each context attribute:

$$C_{Lc} = (Sc_1, Sc_2, Sc_3, \dots\dots, Sc_n)$$

- **Context Property (C_P):** Context Property is a leeway of a Context Lacuna. Each scope Sc_i ($1 \leq i \leq n$) is a C_P
- **Context Plight (C_{Pt}):** Context Plight at time 't' represents particular states in which it is present:

$$C^t_{Pt} = (C^t_1, C^t_2, \dots\dots..)\ C^t_i \in Sc_i\ (1 \leq i \leq n)$$

- **Context Whereabout (C_{Wb}):** It is a scenario in which the end user is interested in above context aware concepts. A smart hospital/context aware hospital can have "C_{Lc} = {Temp = [low, medium, high], Pressure = [>threshold, ≤threshold], location = [living room, bed room, cooking room, wash room]}" as Context Lacuna. "C_P = {temp, pressure, location}" as Context Property. "$C_{pt}^{5:00}$ = {medium, >threshold, living room}" as Context Plight and context Whereabout as:

- $C^1_{wb} = \{q_0, q_1, q_3\} \cup \{\lambda_1, s_2\} \rightarrow$ On entering L.R, T.V switched on
- $C^2_{wb} = \{q_0, q_1, q_2, q_6, q_{10}, q_{13}, \ldots\ldots, q_{11}\} \cup \{\lambda_1, s_1, \lambda_4, \lambda_{10}, s_3, \lambda_{12}, \ldots\ldots, s_{7/8/9}\} \rightarrow$ While resting, lights turned off and A.C turned on (can regulate-low/medium/high)
- …… ensures that when A.C is turned on, windows and doors are closed
- $C^3_{wb} = \{q_0, q_1, q_2, q_5, q_9, q_{12}\} \cup \{\lambda_1, s_1, \lambda_3, \lambda_9, \lambda_{13}\} \rightarrow$ On entering kitchen, exhaust fan rotates when using electric stove
- $C^4_{wb} = \{q_0, q_1, q_2, q_4, q_8\} \cup \{\lambda_1, s_1, \lambda_2, \lambda_6\} \rightarrow$ On entering wash room, lights turned on

The above mathematical notations are able to capture the context/situation in an environment precisely. Using these, it is possible to represent several kinds of context-related parameters formally for any smarter environments. These notations can be further formalized in synchronization with other modeling languages (textual as well as visual (graphical)) so that next-generation context-awareness can be concisely and unambiguously represented, persisted and exchanged.

Mathematical theatrics for the context-awareness properties: A non-deterministic finite state automaton or acceptor non-deterministic finite state machine is quintuple (Σ, S, Q, q_0, δ, F, X, K) in which Σ is the input alphabet taken from non-empty finite set of symbols $\{\lambda_1, \lambda_2, \ldots\ldots\ldots, \lambda_{16}\} \cup \{L_0, L_1, L_2, \ldots\ldots, L_7\}$, S is the switching element with s = {0, 1} indicating the two states whether it is switched (1) or not (0), $Q = \{q_0, q_1, \ldots\ldots\ldots, q_{18}\}-\{q_{15}\} \cup \{L, M, H\}$ is a non-empty set of finite states with the state space R^n and with q_0 as the initial state, δ is the state transition function which is defined as:

$$\delta = Q\times\Sigma\rightarrow\in(q)$$

that returns set of states, $F\subseteq Q$ with set of final states from the set $\{q_3, q_8, q_{12}\}\cup\{L, M, H\}$, X is the set of exceptions $\{E_1, E_2, E_3, \ldots\ldots\ldots, E_7\}$ and finally with set of key constraints represented by $\{C_1, C_2, \ldots\ldots, C_5\}$.

Now, <Q, E> is a directed, controlled context aware automated graph. R^n be the automaton's state space. We can have a stub with its definition as:

- **Hop:** $R^n \rightleftharpoons R^n$ showing that the state changes when traversing along edge $e\subseteq E$ with $(q_0; x_0) \in Q\times R^n$ as inceptive situation

The transition relation: $(q; x)\rightarrow(q^1; x^1)$ is possible if and only if the state of entity is in the domain of hop/jmp.

Emboldening context-awareness concepts: There are sensing, perception, vision, decision-enabling and actuation technologies flourishing to create and sustain context-aware environments. People get excited when they get all kinds of information, commercial transaction, knowledge and physical services based on their situation (location, time, etc.) and their various needs (mental, physical, social, etc.). There are cyber applications, cloud-based services, user devices,

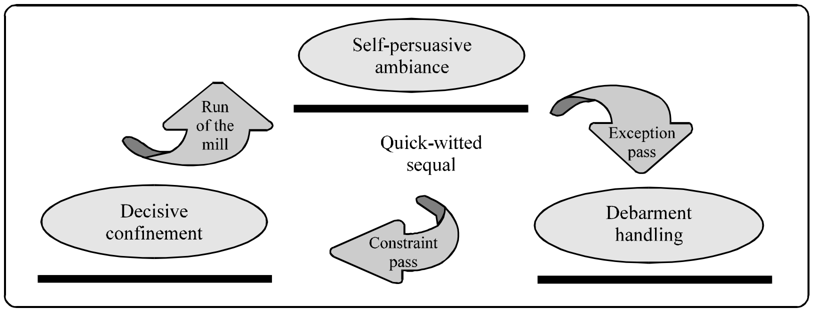

Fig. 2: Quick-Witted Sequel

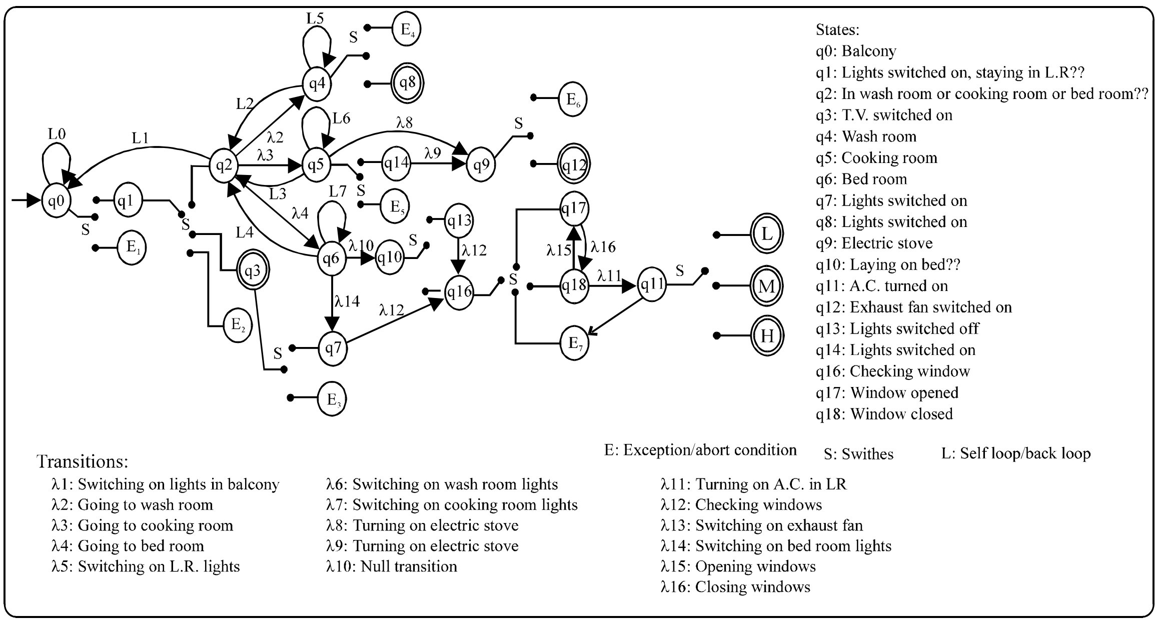

Fig. 3: Smart Homes-the automata representation

communication gateways and device middleware, digitalized and interconnected objects, etc. collectively contributing for the swift implementation and delivery of cognitive context-aware services to users. In this study, we came up with a new idea for having an intellectual self-persuasive environment. A "Quick-Witted Sequel" is a peculiar and an earmarked architecture which makes the environment self-aware containing three main areas self-persuasive ambiance, debarment handling and decisive confinement as presented in Fig. 2:

- **Self-persuasive ambience:** This ambiance indicates a smart environment in the form of an automaton which is non-deterministic and finite that contains all the chunks in a smarter home/context aware home as its states, with the transitions taking place as activities, along with switches (that helps switching from one state to that of an alternative) and exceptions (indicating debarments) as given in Fig. 3 as Smart Homes-the automata representation.

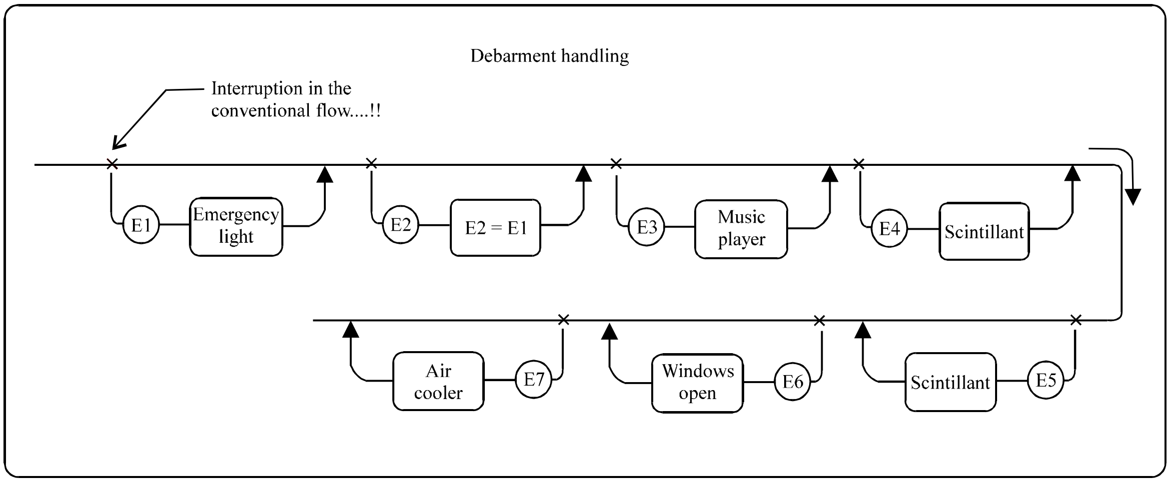

Fig. 4: Exception handling in context-aware ambiance

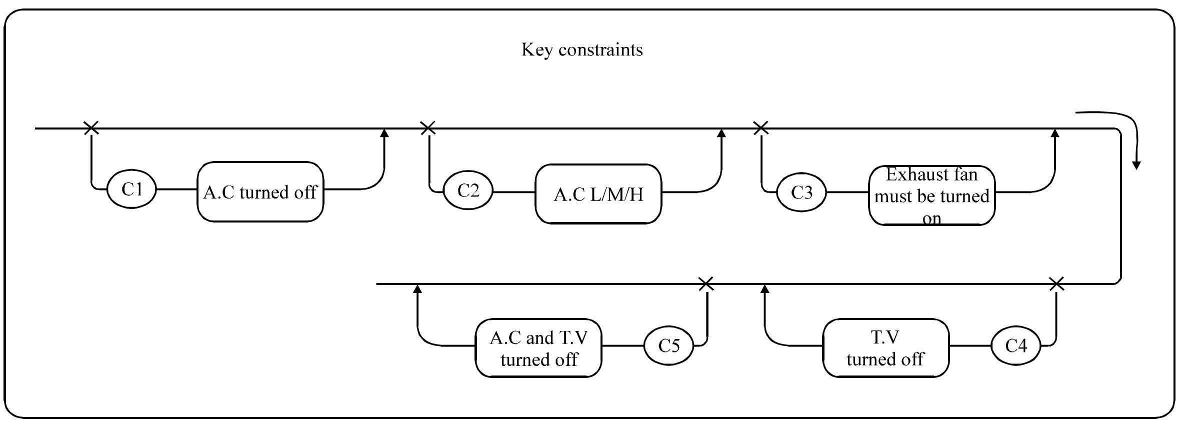

Fig. 5: Policy-based home automation

- **Debarment handling:** As shown in the Fig. 4, the exceptional cases are represented as 'E'. Hence, here in this debarment handling, those exceptions that are caught in the ambiance have to be handled in a conciliatory manner

  Handling exceptions:

- <Lights >→<Emergency lights>
- <Television >→<Music player>
- <Bed light >→<Scintillant>
- <Air conditioner>→<Air cooler>

- **Decisive confinement:** The key constraints for smart automaton presented in Fig. 5 as policy-based home automation is as follows:

C1: If suppose A.C is turned on for more than 2 h continuously, then turn it off for some time say 10-15 min and then turn it on again.

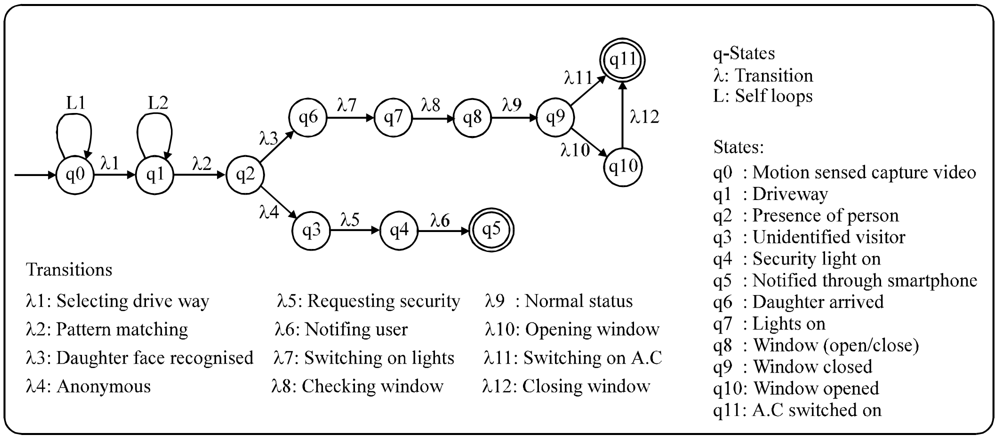

Fig. 6: Automaton for context-aware security

C2: If R.T<OPT, A.C must run in LOW mode, else if R.T = OPT, A.C must run in medium mode else A.C must run in HIGH mode where, R.T: Room temperature, OPT: Optimum temperature.

C3: Don't turn off the exhaust fan if the electric stove is in ON state.

C4: If P.C>A.C, turn off television where, P.C: Power consumption, A.P: Available power.

C5: If P.C>>A.P, turn off both television and air conditioner where, P.C: Power consumption, A.P: Available power.

A sample context-awareness through automata representation: There are a growing array of interesting use cases for context-awareness in our personal as well as professional environments. Here is the one presented in the Fig. 6. The data collected from a security/surveillance camera will be able to proactively determine if the person parking her car in your driveway is your daughter or not. If so, the energy management solution will turn on the air conditioning in the house and some interior and exterior lights. If not, the energy management solution will turn on exterior security lighting and notify your smartphone that an unidentified visitor is in your driveway.

It is clear now that there is a beneficial synergy between the automata theory and context-awareness for establishing smarter environments. Most of the context-related properties are unambiguously denoted by automata diagrams. Any kind of sophisticated and complicated scenarios and situations can be chicly denoted by automata models.

A service-oriented framework for real-time context-awareness: Service Oriented Architecture (SOA) (MacLennan and Van Belle, 2014; Islam *et al*., 2013a) has been a thriving architectural pattern for building multi-tiered applications. That is, all kinds of highly interrelated and complex applications are being constructed as a dynamic collection of interoperable, composable and reusable services. There is no iota of doubt on the statement that context-aware applications will be the most prominent and dominant ones in the days to come. However, developing context-aware applications is not an easy job. As the number of participants and contributors is steadily growing up, the inherent difficulty in forming next-generation situation-aware systems

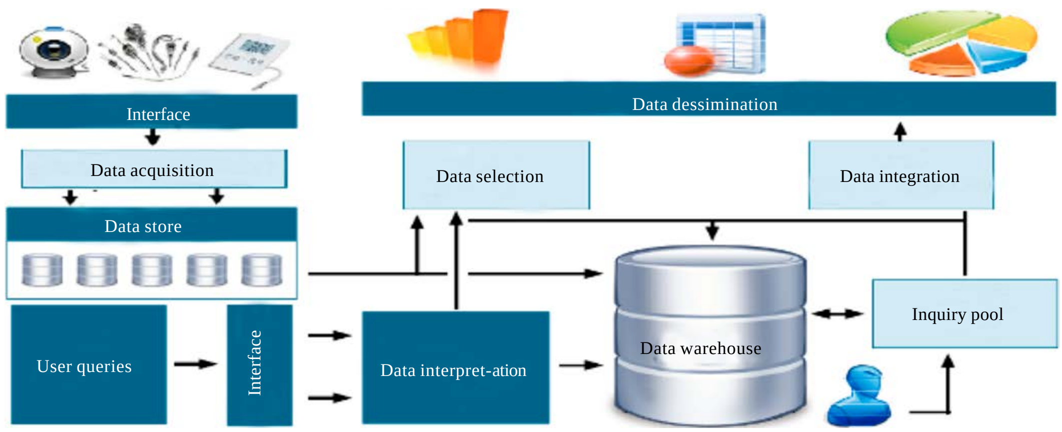

Fig. 7: An enabling context aware framewor

is widely accepted. Having understood this perpetual need, we have adeptly leveraged the shining SOA concepts in designing and implementing a viable and value-adding framework for context-aware computing.

Let us consider an environment wherein there are 'n' devices and sensors (hereafter termed as participants). A software infrastructure which is a part of the framework as given in Fig. 7 is designed in such a manner that helps in collecting the multi formatted data emitted from those active participants and in stockpiling it safely into any data store. This is nothing but a "DATA ACQUISTION" phase which is prior and most vital in any design. Now, a user interface lets our users to provide their queries in a natural language with certain constraints posed on it. The backend processes user's query and selects the desired data from the database thus naming it as "DATA SELECTION". Now, the next phase "DATA INTEGRATION" deals with the integration of multi formatted data thereby transforming it into a universal single accepted representation that helps in decision making. The hindmost phase "DATA DESSIMINATION" of this design is rendering the prepared and refined data in a format that can be easily understood by a common user without any knowledge regarding the backend processing. Also an auxiliary way is provided to the users to get the details of any threats to the home such as possibility of fire, short circuit etc.

The ultimate output of the enabling context aware framework is the context/situation details in a preferred format. The context information can then be forwarded to the authenticated target devices in order to contemplate appropriate counter measures and activate processes to initiate the desired activities.

Context engineering framework steps: The framework involves several steps that are automated through a cohesive and loosely coupled software components. This section vividly illustrates the different phases.

Data acquisition: All the disparate and distributed participants capture all kinds of data about systems, their states and the state changes and pass on them to target systems to proceed with the next set of activities. The peripherals of data acquisition structure encompasses:

- Participants that makes physical parameters as electrical signals
- A conditioning circuitry that makes the above sensor signals as digital values

Data integration and transformation: As data being received from diverse and geographically distributed participants are heterogeneous in nature, there is a need for automated toolkits for capturing, polishing, filtering, cleansing and facilitating data translation and formatting so that the next-in-line data interpretation engine can run smoothly to extract all kinds of hidden data-driven insights.

Data interpretation: This is all about the extraction of useful patterns, tips, associations, opportunities, risks, etc., so that both tactic as well as strategically sound intelligence can be passed on to users and decision-makers to ponder about the next course of action with all alacrity and authentication. Policies are being provided to data fusion engine in order to take them into consideration while coming out with actionable insights for both men as well as machines. Machine-to-machine (M2M) integration capabilities empower all sorts of electronics to bring in right and relevant automation, acceleration and augmentation.

Data selection: This module/phase of the framework allows the user to select the data according to their interest. That is the user provides a query through an interface to the database and that, in turn, results the data in which the user is interested in.

Data dissemination: This is the final module in the lifecycle. That is, insights need to be presented to authenticated and authorized users and devices in their preferred format. There are data visualization tools and a variety of dashboards for illustrating graphs, charts, maps, etc.

RESULTS AND DISCUSSION

Framework implementation descriptions: The proposed method implemented the full framework and used it in order to substantiate the idea of context-awareness in an effective and efficient manner. The system configuration details are given:

- Quad-core processor of 2.2 GHz
- 8 GB RAM storage
- 500 GB disk storage
- Java 7 is installed and the environment variables are appropriately added

The phpMyAdmin is a free software tool written in PHP, intended to handle the administration of MySQL over the Web. Java language is the primary language leveraged for the framework implementation. The screenshots given below show a smart home implementation that simulates a splash of events and actions that arises in daily life of our inland. As shown in the Fig. 8, the values from different sensors embedded in a smart home get updated every time in a database.

An interface is provided to the sensors that help in capturing the data time to time. Figure 9 shows the interface between sensors and the data storage. This interface helps in capturing the data from sensors and successfully stores it in a database for future references and categorization of the collected data (data interpretation) presented in Fig. 10 and data dissemination is given in Fig. 11 and corresponding user interface for queries regarding any threats (inquiry pool) is given in Fig. 12.

Algorithms used for data capturing and interpretation modules: A set $S = \{s_1, s_2, s_3, \ldots\ldots\ldots, s_n\}$ represents the sensors embedded in a context aware home.

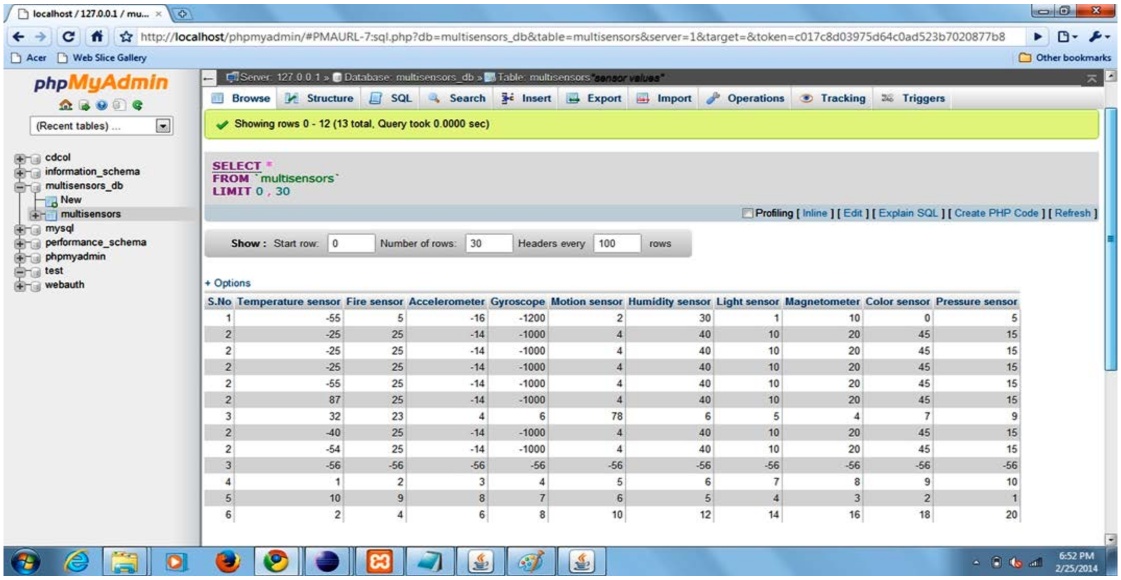

Fig. 8: Values from sensors stored in a database

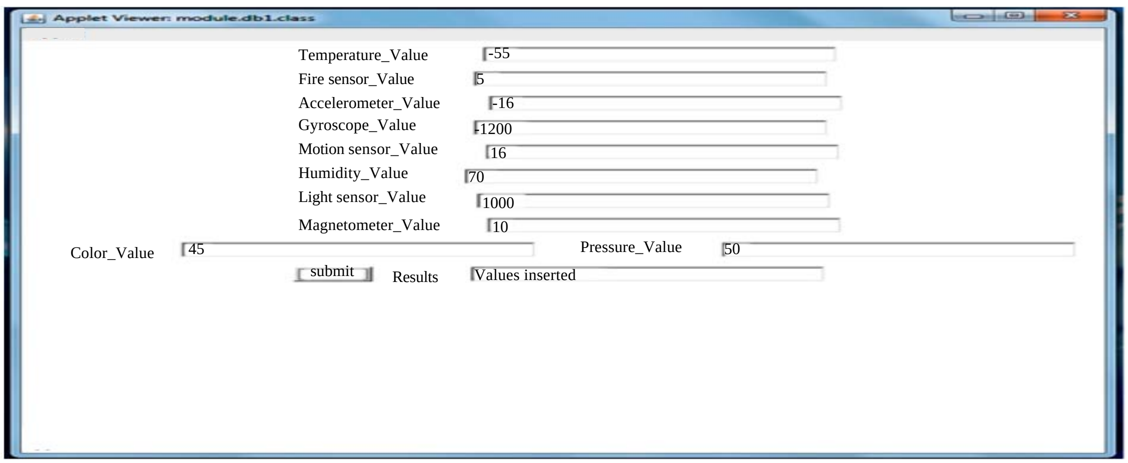

Fig. 9: Sensor-data store interface (acquisition)

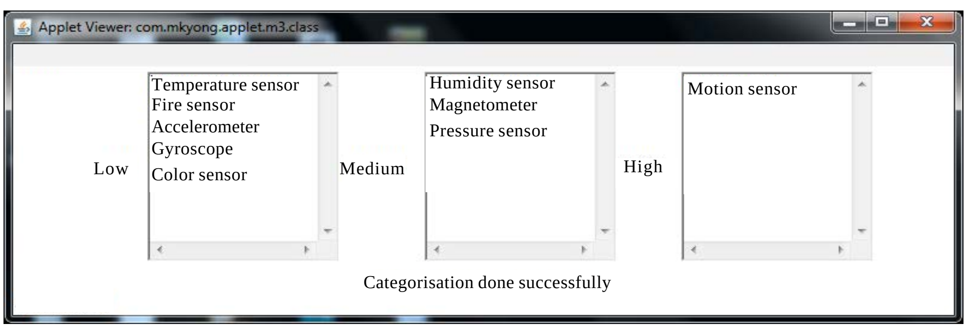

Fig. 10: Categorization of the collected data (data interpretation)

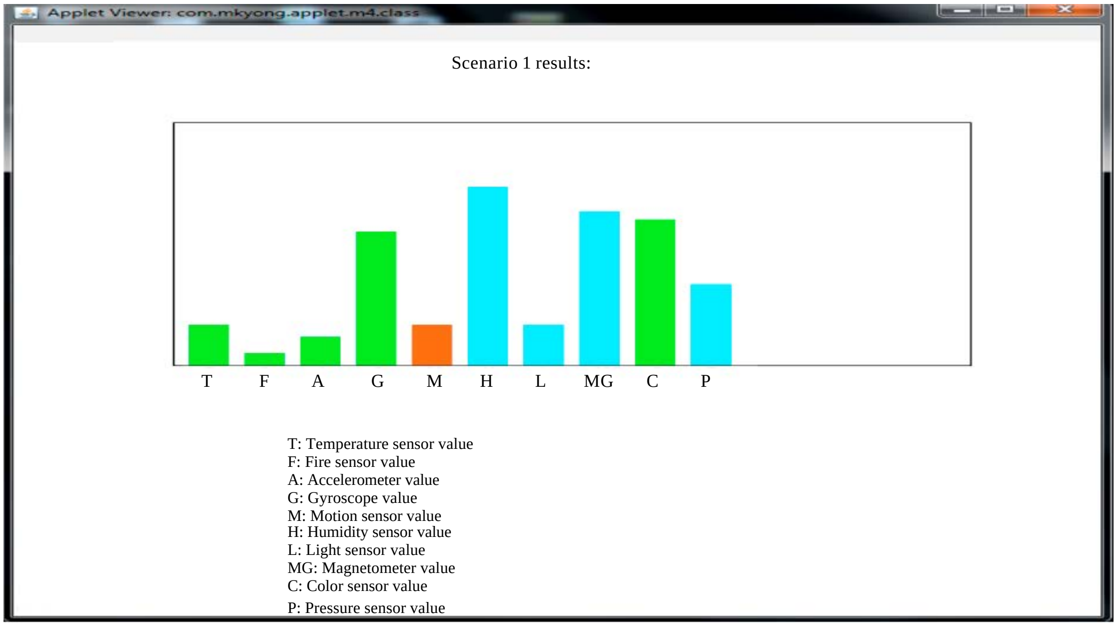

Fig. 11: Data dissemination

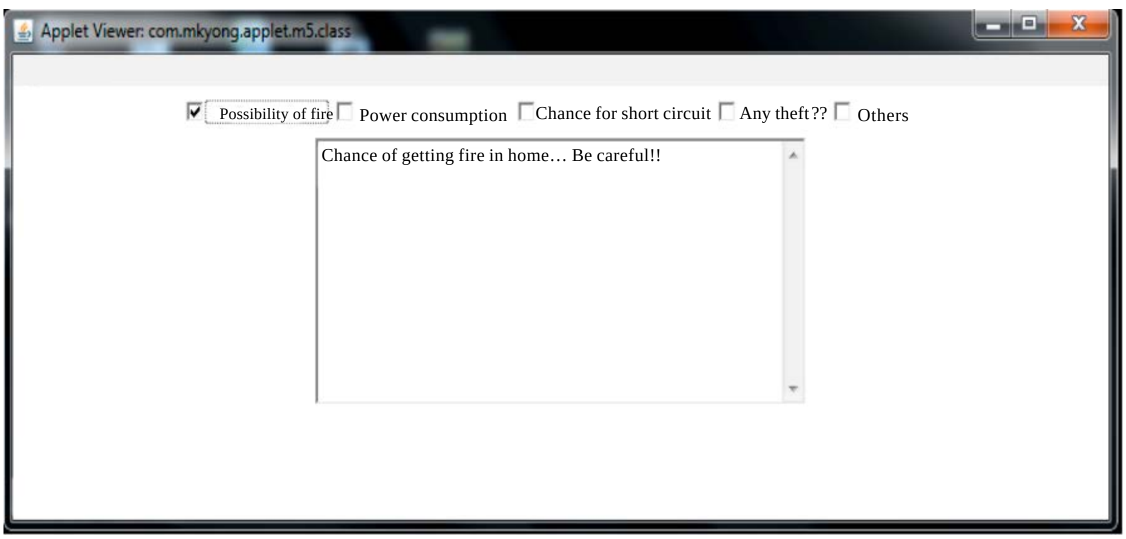

Fig. 12: User interface for queries regarding any threats (inquiry pool)

Different sets, each representing set of sensor data at a particular time gets automatically updated to a database. Hence, S_{t1}, S_{t2}, S_{t3}, ... , S_{tn} are the data sets that got updated in a database at different time intervals:

$$\{s_1 \in S_{t1} \mid s_1 \text{ recorded at } t_1\};\ S_{t1} \neq \emptyset \text{ also } \# S_{t1} = n \mid n>0 \text{ [representing cardinality]}$$

$$\{s_2 \in S_{t2} \mid s_2 \text{ recorded at } t_2\} \ldots\ldots\ldots\ldots\ldots \{s_n \in S_{tn} \mid s_n \text{ recorded at } t_n\}$$

Algorithm involved in categorization:

```
X and Y are the initial sets with
IP: X = {s1, s2, ..... sn } and Y = ϕ
OP: Categorized sets
Start:
X = {x∈s | S(x)}
      While (not the end of set)
            if(x | x>threshold)
                  Y = {x∈X | S(x)}
            else
                  No changes to the set
end loop;
end;
```

Integration function: From the set S into the set P, the set of functions is the set S→P. The signature as an operator on sets, under the interpretation of (→):

$$f: S\rightarrow P \text{ is a member of the set } S\rightarrow P \mid f\in S\rightarrow P$$

Defining the function as:

$$f(x) = (x\times 100)/\text{Threshold};\ x\in S$$

Analyzing user's request:

- S1 and S2 | S1⊂S and S2⊂S
- D = S1×S2 = {(x, y): x∈S1 and y∈S2}
- E.g.: If S1 = {Temp_sensor, Smoke_sensor} and S2 = {Fire_sensor}

  Then, D = S1×S2 = {(Temp_sensor, Fire_sensor), (Smoke_sensor, Fire_sensor)}:

- To analyze user query "Possibility of fire"... The elements in the set D are used
- If both the Temp_sensor and Fire_sensor crosses the threshold, then fire accident may occur in the house

Advantages: This framework has intrinsically used a bevy of proven algorithms for various steps ranging from data collection to knowledge engineering. This is a web-based application so that anyone with the internet connectivity can use this system for making his environment smart. This is being polished further in order to work in smartphones (Android, iOS, BlackBerry and Windows Phone) not only in mobile phones but also in all kinds of electronic devices (wearable, portables, fixed, etc.). With the latest phones are being stuffed with a variety of sensors, context-awareness can be being speeded up and simplified through this framework application. That is, all the environmental sensors are in direct interactions with sensors attached with phones to enable phones to act as the most prevalent and popular gateway for context-awareness for people in different locations in their daily lives.

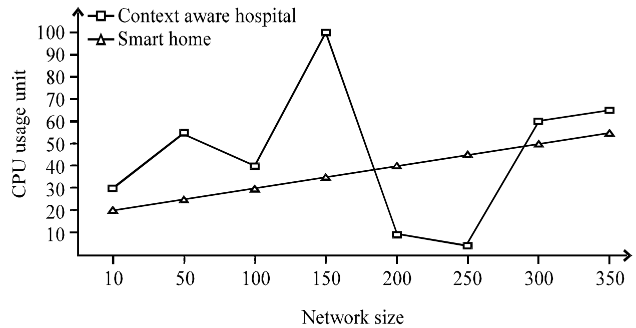

Fig. 13: CPU usage of smart homes and context-aware hospital

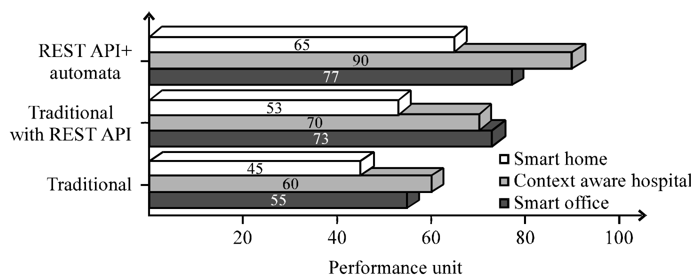

Fig. 14: Performance evaluation of the traditional approach vs. REST API+Automaton approach in context-aware hospital

Performance evaluation: In the CPU usage of two sample context-aware applications; smart homes and context-aware hospital is given in Fig. 13.

The performance evaluation of the Traditional Approach (TA) and the new approach introduced by us using the REST API+Automaton is indicated in Fig. 14. The performance of our method is comparatively better as per the diagram.

The thorough analysis is made among the methods that implement context-aware computing as mentioned in Makris *et al*. (2013). This is presented in the bar graph (Fig. 14). Where comparison is depicted among the Traditional Approach (TA), the TA with REST API and the third approach (TA with REST API and Automata) (Hatzi *et al*., 2012). The performance shown is far better for different use cases (Smart home, smart office and Context-aware hospital) as suggested in Zhang *et al*. (2013).

CONCLUSION

Context-awareness is being touted as the prime capability for any futuristic system and service to be decisively adaptive in their operations and outputs. However, gathering context information and feeding them to all kinds of electronic and digitalized devices in our daily environments is facing a lot of specific barriers and blockages. Scientists and scholars across the globe, therefore, are showing exemplary interests in unearthing robust and resilient solutions to overcome all these prickling and perpetual issues with the vision of enabling our everyday artifacts smart. Having taken all the technical advancements in the happening field of IT into considerations, we have come out with an easy-to-use implementation framework that is capable of gleaning all the right and relevant situational and decision-enabling information in time to devices in our personal as well as professional environments with all the right and relevant knowledge to empower them to act as per the changing needs. Service Oriented Architecture (SOA) and Event-Driven Architecture (EDA) are the widely deliberated and asserted architectural patterns for enterprise computing. We have done the necessary tweaking on them to come out with the framework for simplifying and streamlining the context-aware computing.

REFERENCES

Abowd, G.D., A.K. Dey, P.J. Brown, N. Davies, M. Smith and P. Steggles, 1999. Towards a better understanding of context and context-awareness. Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing, September 27-29, 1999, Karlsruhe, Germany, pp: 304-307.

Albreshne, A., A.A. Lahcen, J. Pasquier and A. Abdaladhem, 2013. A Framework and its associated process-oriented domain specific language for managing smart residential environments. Int. J. Smart Home, 7: 377-392.

Bandyopadhyay, S., M. Sengupta, S. Maiti and S. Dutta, 2011. A Survey of Middleware for Internet of Things. In: Recent Trends in Wireless and Mobile Networks, Ozcan, A., J. Zizka and D. Nagamalai (Eds.). Springer, Berlin, Heidelberg, ISBN-13: 9783642219368, pp: 288-296.

Bettini, C., O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan and D. Riboni, 2010. A survey of context modelling and reasoning techniques. Pervasive Mobile Comput., 6: 161-180.

Chen, G. and D. Kotz, 2000. A survey of context-aware mobile computing research. Computer Science Technical Report No. TR2000-381, Dartmouth College, Hanover, NH., USA. http://www.cs.dartmouth.edu/reports/TR2000-381.pdf.

Hatzi, O., D. Vrakas, M. Nikolaidou, N. Bassiliades, D. Anagnostopoulos and L. Vlahavas, 2012. An integrated approach to automated semantic web service composition through planning. IEEE Trans. Serv. Comput., 5: 319-332.

Hong, Z., P. Li and W. Jingxiao, 2013. Context-aware scheduling algorithm in smart home system. China Commun., 10: 155-164.

Islam, M.A., F. Belqasmi, R. Glitho and F. Khendek, 2013a. The design and implementation of OMA restful location services in wireless sensor environments. IEEE Commun. Mag., 51: 122-131.

Islam, M.M., J.H. Lee and E.N. Huh, 2013b. An efficient model for smart home by the virtualization of wireless sensor network. Int. J. Distrib. Sensor Networks, Vol. 2013 10.1155/2013/168735

Lee, C.S., G.M. Lee and W.S. Rhee, 2014. Smart Ubiquitous Networks for future telecommunication environments. Comput. Standards Interf., 36: 412-422.

Lee, H.J., Y.S. Son, J.H. Park, K.D. Moon and J.C. Ryou, 2012. General middleware bridge to support device interoperability on different middlewares. J. Inform. Process., 20: 525-530.

Liang, Q., B.S. Lee and P.C. Hung, 2014. A rule-based approach for availability of service by automated service substitution. Software: Pract. Exp., 44: 47-76.

Lopes, F., F.C. Delicato, T. Batista, E. Cavalcante and T. Pereira *et al*., 2014. OpenCOPI: Middleware integration for ubiquitous computing. Int. J. Parallel Emerg. Distrib. Syst., 29: 178-212.

MacLennan, E. and J.P. Van Belle, 2014. Factors affecting the organizational adoption of Service-Oriented Architecture (SOA). Inform. Syst. e-Bus. Manage., 12: 71-100.

Makris, P., D.N. Skoutas and C. Skianis, 2013. A survey on context-aware mobile and wireless networking: On networking and computing environments integration. IEEE Commun. Surveys Tutorials, 15: 362-386.

Molla, M.M. and S.I. Ahamed, 2006. A survey of middleware for sensor network and challenges. Proceedings of the International Conference on Parallel Processing Workshops, August 14-18, 2006, Columbus, OH., USA., pp: 223-228.

Ntanos, C., C. Botsikas, G. Rovis, P. Kakavas and D. Askounis, 2014. A context awareness framework for cross-platform distributed applications. J. Syst. Software, 88: 138-146.

Strang, T. and C. Linnhoff-Popien, 2004. A context modeling survey. Proceedings of the 1st International Workshop on Advanced Context Modelling, Reasoning and Management at UbiComp, September 7, 2004, Nottingham, UK.

Warriach, E.U., 2013. State of the art: Embedded middleware platform for a smart home. Int. J. Smart Home, 7: 275-294.

Zhang, D., H. Huang, C.F. Lai, X. Liang, Q. Zou and M. Guo, 2013. Survey on context-awareness in ubiquitous media. Multimedia Tools Applic., 67: 179-211.