Comprehensive Study on Context Aware Computing Modeling Techniques

K.S.Sagaya Priya*, Dr.Y.Kalpana**

*Research Scholar, Dept. of Computing Science, Vels University, Chennai, Tamil Nadu, India.
**Associate Professor, Dept. of Computing Science, Vels University, Chennai, Tamil Nadu, India.

Abstract- The context-aware system is one of a significant area of ubiquitous computing, by means of the expansion and appearance of mobile devices, where context-aware applications are designed for performing the stable modifications in the cloud environment. The context-aware system is a key parameter that has generated different approaches and heterogeneity of the different domains for context modeling. With ubiquitous computing is developing more popular at the present time and the user and his tasks are becoming the focus of application improvement. The area of context-aware computing has developed within the support of different kinds of context models and reasoning techniques. Several existing approaches for modeling context information and every model brings along various reasoning techniques. In this paper, the terms of the context and context-awareness, discuss about the significance of context-aware computing, the need of context modeling and discusses some context modeling.

Keywords- Context aware computing, context modeling, Ontology based models, Logic based modeling

I. INTRODUCTION

With the appearance of mobile devices, ubiquitous systems have gained attention and application developers have progressively more responsible on creating applications that focus on smart phones, notebooks, PDAs, tablets etc and that the way to use these devices and new technologies to assist the user for performing its daily tasks. In this process has resulted in speedy integration of these devices in a person’s to utilize everyday life that the user in a convenient manner. In the forepart and central part, these tasks have been placed for improving service development, suppressing the devices, their connectivity and other technical problems.

Particular field of pervasive computing is perspective awareness, an idea to introduce by Schilit and Theimer in 1994 referring to a new class of applications that are alert to the context where they are executed. Focusing on active map service that provides information about the located objects and how they modify at the time, they use the term context to refer to location and that they understand context-awareness because the ability of mobile applications to find and react to changes of the environment wherever they are located. [1] Schilit later provides a user-centered definition of context stating that context can be characterized by many important aspects: where are you from, who are you with, and what resources are require nearby. [2] According to the information about the location is simply as a segment of the context information and context is conferred as a relation between the user and also the environment.

Consequently, many other researchers and scientists have continued working within the domain, increasing and redefining the thought of context. Two distinct approaches of context awareness for human-computer interaction and its additional development progress following two ways. While referring to completely different context parameters are being given different preference. Context parameters that application developers and researchers usually list are location, time, environmental parameters, user activity, device capabilities, identity, network capability etc. Application developers focus into the ‘when’, ‘where’, ‘what’ and ‘who’, and of certain entities and by its analysis they reason about the ‘why’ of a given occurrence, and program the application logic.

Schilit et al. [3] have also before discussed about context awareness as the capability of a system to “adapt consistent with the location of use, the gathering of nearby people, hosts, accessible devices, moreover on changes to such things over time”. An application with these abilities can be able to sense the environment and correspondingly react to its changes. In Table 1 illustrate the outline of four categories in the context-aware applications. The categories are divided according to two properties: whether or not the system gathers information or executes an action, and whether this is often done manually or automatically.

<table>
<thead>
<tr>
<th>Information</th>
<th>Manual</th>
<th>Automatic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proximate selection and context information</td>
<td>Automatic contextual reconfiguration</td>
<td></td>
</tr>
<tr>
<td>Command</td>
<td>Contextual commands</td>
<td>Context – triggered actions</td>
</tr>
</tbody>
</table>

Proximate selection is considered as applications when manually request information depend on current disposable context, while automatic contextual reconfiguration category encompasses applications that automatically retrieve information based on this context. Contextual command applications are requested to execute commands for the user when they are instructed manually to do process and context awareness.
triggered applications refer to applications that provide automatic execution of commands depending on available context.

1.1 Context Categorization

Classification of context aware system is really supportive to understand, uncover, manipulate and sort out a variety of contexts in an efficient manner. Moreover, it can provide huge support for users to identify the sort of a given context before using it. As context could be classified from the completely different perspective concept, given below justify the different context categorization [4].

- **5W1H (Who, When, Where, What, Why and How):** At this point of view to analysis the environmental situations and intuitive to understand. For example: **Life assistance of elderly**, Sen Peter, 8 o’clock, backyard, picking fruit, ripe apples, by way of a ladder.

- **Physical/Virtual:** Differentiation confers to some sources such as sensing devices (physical); user, context servers, etc. (virtual). It is simple other than ambiguity to identify the same context as it can be physical or virtual depending on different conditions. Example: **Rehabilitation, Physical:** heart rate **Virtual:** patient’s medical history from the database.

- **Static/Dynamic:** Observation over time means that always equal (static) or adaptive to modify in the environmental situation (dynamic). Intuitive to understand. For example, **Plant inspection, Static manner:** the place where a tree grows in nature Dynamic manner: Due to this aspect of the tree in the current season.

- **Direct/Indirect:** For example, **Birthday, Direct context:** the real date is the birthday **Indirect context:** in which birthday is it and does this mean something (e.g., 60th Birthday). Indirect context is more complex to acquire and differentiation refers to obtainment complexity, needs computation, inference etc. trouble-free to recognize

- **Navigation Sensed, Combined, Inferred and Learned:** In advance differentiation of obtainment complexity by means of sub-categorizations. For example: **Navigation Sensed:** analysis an object proximity **Combined:** rapid process and direction of motion **Inferred:** check distance (rules) **Learned:** compare with similar situations.

- **Internal/External:** At this point of view to differentiate sources from the users. The same context, ambiguity handles between two or more situations. Example: **Life assistance, Internal:** desire to get up from the bed **External:** it is the time to get up.

- **Primary/Secondary:** For example: **Health monitoring, primary:** check the blood pressure **Secondary:** comparison of historical blood pressure data. Obtainment complexity simply close to direct / indirect and then ambiguity in identifying the complexity of obtainment.

1.2 Context Lifecycle

Even though current proposals for context-aware middleware contain totally different components or modules to manage context, they adapt a general rule of the context lifecycle. The life cycle of context is that the period of time from its obtainment to destruction. Context lifecycle differentiates six significant events as Context Acquisition, Context Modeling, Context Reasoning, Context Distribution, Context Repository, and Context visualization as the sequence of process shown in Figure 1.

The context awareness begins with the acquisition of different types of context followed by the formalization and inference process. Finally, ends up with the context distribution to the corresponding applications. At the point of context modeling and reasoning, historical context data requires to be recorded for additional use or queries and also can be visualized by users. The major phases of the lifecycle are outlined thoroughly.

**Figure 1:** General context lifecycle.

1. **Context Acquisition:** The main aim of context acquisition is able to achieve a maximum amount of data, so that the possibilities for applications to be intelligent could be maximized due to more affluent context information.

2. **Context Modeling:** In the acquisition phase, a massive amount of context data that is obtained in multiple formats. The premise is to define and store it in an exceedingly machine-readable and process form, herewith all data should be converted into a unique format such that the context can be understood and shared. A model represents processes and describes, the object “context”. In different surveys, on popular context modeling methods is published like, “User-centric” “Object-oriented”, “Key-value”, “Markup”, “Graphical”, “Ontology-based”, “Logic-based”, “Chemistry-inspired”, “Multidisciplinary” and “Domain-focused”.

3. **Context Reasoning:** Context reasoning is also known as inference and derives on-demand service from the fact that context data is imperfect and uncertain naturally [5]. The task of context reasoning is to reduce high-level context from a raw context that has related to a few basic
functionalities performances like checking context inconsistencies, filling in missing values, removing outliers, validating the context values, and applying a few calculations to obtain new values.

4. **Context Distribution:** Context distribution is completely responsible for disseminating helpful context information to corresponding applications. Two typical distribution mechanisms (*subscribe/publish and polling*) are widely utilized in current solutions [6]. It is also referred to as Notification like Subscribe/Publish. Applications curious about certain context data can subscribe to the middleware and be notified when updates of the registered context information arise. Polling - Context consumers are capable to actively make queries for their interested context information at every moment. Depending on the user modeling and reasoning techniques, totally different query methods can be employed.

5. **Context Visualization:** Context visualization provides new techniques for viewing data. There is a growing need for visualizing in an effective manner to offer a visual overview, explore, analyze, and present phenomena which are repeatedly difficult to know or imagine. Since ontology-based modeling is selected as the best modeling technique within the context modeling phase, research on context visualization will be targeted on reviewing current approaches to visualize ontology-based context data. In *context visualization*, ontology-based context data is a complex task for the reason that it means to enrich data with hierarchy, relationships, etc. Therefore, context visualization methods can be ontology-tailored visualization techniques or adapted from other techniques such as graph or file system visualization.

### 1.3 Architecture

In general architecture of context-aware systems can be implemented in many ways. This approach based on special needs and conditions such as the location of sensors (remote or local connections), the available resources of the used devices (PCs or smart phones), the number of possible users (one user or many) or a further extension of the system. It predetermines the architectural style of the context aware system at least to some extent. Three completely different approaches represent on how to acquire the contextual information [7].

- **Direct sensor access:** With sensors locally built in this approach is an often utilized using a device and the client software gathers the required data directly from these sensors. Drivers perform for the sensors are hardwired into the application and then tightly coupled method in some rare cases. Hence, it is not suited for distributed systems because of its direct sensor access that lacks a component capable of managing multiple concurrent sensor access.

- **Middleware infrastructure:** With the purpose of hiding low-level sensing details, introduce middleware-based approach for performing with a layered architecture to context-aware systems. The modern software utilizes a few techniques of encapsulation to separate for example, GUI (graphical user interface) and business logic. Middleware infrastructure is compared with the previous sensor access so that technique effortlessness extensibility. As the client code simplifies the reusability of hardware dependent sensing code because of the strict encapsulation and has not been changed anymore.

- **Context server:** This distributed approach extends the middleware based architecture by introducing an access managing remote component control, which is to allow multiple clients access to remote data sources Gathering sensor data is transmitted to context server for facilitating concurrent multiple accesses. The main advantage focus on the relieving clients of resource-intensive operations and reuse of sensor usages. End devices are probably utilized in context-aware systems were mobile gadgets with limitations in power computation, disk space etc., To considers appropriate protocols, quality of service parameters, network performance etc., when designing a context-aware system based on client-server architecture.

![Figure 2: General Architecture](image)

In Figure 2, shows a common conceptual architecture for a context management system to obtain contextual information for context-aware applications. Sensors (it explicitly...
comprises software components) provide raw environmental data, which is developed by the context capturing interface into contextual information. By using software, the raw data is abstracted into separate data structures so that it can be additionally processed. The contextual information is executed into the context repository that is responsible for the persistent storage of the contextual information.

The manual change of the context is encoded by using inference rules that are enforced by the inference engine. Both contextual information and rules are loaded by the inference engine, which subsequently updates the context in the repository corresponding to the inference rules. The inference engine could also trigger any actuators that are affected by the context update. The context API provides access to the context for user applications, and it performs the architectural cut between context utilization and context management.

1.4 Sensing Infrastructure
To enhance the reusability of applications and the process of building them, a common practice is to separate the sensing logic from the rest of the system. Thus a common basic module for all architectures is that the sensing layer. However, the term does not only infer hardware sensors, and sources of contextual data; it encompasses any source which provides contextual data and enhances the description of a real situation. Three types of sensors are classified physical, virtual and logical sensor that can be identified depending on the way the data is measured and acquired. [8] Sensors can be classified into three teams (Indulska and Sutton, 2003).

- Physical sensors: The most regularly utilized this kind of sensors is physical sensors. Nowadays, several hardware sensors are ready to accessible and capable of gathering any physical data. For example of physical sensors such as color sensors, IR, Photodiodes, and ultraviolet sensors, motion detectors, accelerometers, various cameras, magnetic fields, Outdoor: global Positioning System (GPS), global System for Mobile Communications (GSM); Indoor: Active Badge system, etc.
- Virtual sensors: Virtual sensors source provides context data from software applications or services. As an example, it is capable to determine an employee’s location not only by using tracking systems (physical sensors) and also detect through a virtual sensor, e.g., by browsing a travel-booking system, an electronic calendar, emails etc., for location information. Through virtual sensors include different context attributes that can be sensed, for example, the user’s activity by verifying the mouse-movement and keyboard input.
- Logical sensors: Combination of physical and virtual sensor provide with additional information from databases or various other sources so as to solve higher tasks in an efficient way, these sensors make use of a couple of information sources. For example, a logical sensor utilize to detect an employee’s current position by analyzing login process at desktop PCs and a database mapping of devices to location information.

II. CONTEXT MODELING APPROACHES
A context modeling is necessary to interpret and sense dynamic context representations at a high-level abstraction in an unobtrusive manner. It has related to variant context sources, accurate representation of context with a high certainty under different conditions of measuring range and sampling methods is very significant to assure the quality of contextual information. To be able to perform, to consider heterogeneity (i.e., imperfectionist dynamic nature), mobility (i.e., asynchronous, timeless data capture) and comparability (i.e., coexistence of comparable context from different sources) of a large variety of context sources at any level of abstraction. Moreover, it considers dependencies and relationships among semantic entities like accuracy and battery computation power. Many context modeling schemes have been proposed [9], [10], [11] given below explain few context modeling schemes.

- Multidimensional context modeling approach: This approach was one of the first approaches proposed in [12] for generalized context modeling. The concept of this approach is to make a decision to the context situation of the entity that is the most similar or relevant. In this approach, completely different context situations are represented as individual examples in the multidimensional area. The model classifies the entities based on their similarities to these examples [13]. The similarity is a decreasing function between the entities and individual examples in a space [13]. The Context space Model (CSM) [16] and Vector space Model (VSM) [14] are the classical examples of these models. CSM operates on these two ideas from geometrical spaces: context state and situation spaces. The context state deals with the current state of the entity being modeled at a convenient time based on the contextual information, while the situation space represents a real-life situation depend on a collection of context states during a certain period of time. This modeling approach has shown practical usage examples in improving context-aware mobile applications [14] and detecting anomalous behavior in video surveillance [15].
- Hierarchical Representation: In this approach represents context utilizing a tree-like structure of context types. Dey et al. [17] used this approach to represent hierarchical representation of context to improve the efficiency of searching in context container. Four essential characteristics of context information were initiated, i.e., identity, location, status (or activity), and time. Identity referred to the ability to assign a unique identifier to an entity, although location was need more than position information in a two-dimensional space. Status (or
activity) identified intrinsic characteristics of the entity that may be perceived, while that time was context data help to characterize a situation [17]. The hierarchical representation has limitations of ambiguity and privacy in sensed context and its relationships among the various contexts. In this illustration model desires additional work to manage and maintain the dynamic environment of the context within the real world. The system should be capable to manage the traditional and general context that cannot be sensed directly from the environment.

- **Object-role based or object-oriented modeling approach**: In this modeling approach was approved from the database modeling field [18]. At this point, the context model language, which was depend on Object-role Modeling Language (ORM), was developed to maintain the object-role based context modeling. This approach utilizes the ideas and advantages of the object oriented approach. The superclass characterizes that the abstract context object (e.g., ContextObject) with abstract methods for obtaining the context information (e.g., processData). Subsequently, a piece of context information (e.g., location, temperature and others) is inherited by the abstract ContextObject class (e.g., LocationObject, TemperatureObject, OtherObject), which will be implement the abstract techniques (e.g., processData) [19]. Another example of such approaches are the Object-Relational Database Management Systems (ORDBMS), which are utilized to model context and relationships between different contextual information. The advantage of this approach has utilized the combination of traditional relational and object-oriented concepts to model the context information.

- **Ontology-based models**: In this model widely make use of complex context situations and ontology affords a uniform way for identifying the model’s core concepts, facts and also an arbitrary amount of sub-concepts, which allows for sharing contextual knowledge process and reusable of applications in a pervasive computing system [20]. To increase complexity for certain types of context-aware applications [14]. Because of the existing ontology based models need more changes in the core model so as to customize and adapt it for a specific application domain; their practical applicability is reduced in regard to mobile application development.

- **Domain Focused Representation**: This context modeling is founded on domain specific information representation. Castelli, et al. [21] suggested a data model for expressing facts that deals with information arriving process from different heterogeneous sources. This approach afforded ease of processing, querying and adaptation to context complete and incomplete information. By means of 4-fields were expressed facts (Who, What, Where, When) and consists of W4 atomic units were provided the factual knowledge process. —Whol was the current user (by entering the user explicitly to the system), —Whatf was the current activity being performed, —Wheref was the current activity performed about location information, and —Whenf was the current activity performed (time duration). In these fields —Wsher and —Whatf were obtained by the Global Positioning System (GPS), while —Whenf was provided by Personal Digital Assistant (PDA) other than combination of GPS and PDA. Main access control of this model is considered for performing less expressiveness and less flexibility. This model requires more flexible strategies for processing context distribution and access to develop adaptability and robustness in the system.

- **Spatial context model**: Space model is a significant factor in various context-aware applications and most context definitions point out space as a vital factor. For example, Schilit and Adams describe about three important features as —Where are you from, —Who are you with and —What resources are nearby with [21]. In case, a few context modeling approaches provide location and space for a preferential treatment. Most of the spatial context models perform with fact-based models and control their context information by viewing physical location. Spatial context models can be illustrated all along the tiers of spatial ontologies proposed by A. Frank [22] also proposed and usually cover all 4 tiers for ontology based models of context information. It is useful to distinguish between various implementations of spatial context models and improved in the Nexus project (called Augmented World Model [23]) which is an object-based class hierarchy of context information that assists multi-inheritance. In the Nexus model, the Equator project context model [24] is also represents a typical contextual ontology that communicates to all tiers by means of an OWL class model. Location model is a hierarchical notion of inter-connected between symbolic spaces, like Buildings, Rooms, and Floors. Even though, the ontology also provides coordinate features, Millard et al. [24] states that manages any inference over them using a normal reasoned and they are usually not spatially aware applications.

- **Key Value Models**: The key-value pairs are the simple data structure for modeling contextual information of this model. Key-value modeling approach is normally used in distributed service frameworks. Previously Schilit et al. [4] exploits key-value pairs to that model the context by providing the value of context information (e.g. location information) and also to an application as an environment variable in a perfect sequence. In such frameworks, the services are typically described about a list of simple attributes in a key-value manner, and the employed service discovery procedure controls an accurate matching algorithm on these attributes.
• Logic Based Models: This kind of modeling representation is depend on adding context as facts and extracting contextual information by using rules or expressions. Formalism is tightly coupled to context reasoning approaches. It can also represent the meta-propositional properties, as a result created a hierarchy of context and Meta information. Gray et al. [25] discussed among the formal representation for sensing context information in First-Order predicate logic [26]. Context representation model requires within appropriate ways to describe about interactive system of the functionality process relevant to use the sensed context information. Due to drawback of this model of sensed context information cannot be used by software tools and does not maintain design and documentation process.

2.1 Application Domain of Context-Aware Services

Advance technologies and context awareness computing evolves; the context awareness is applied and increased in the number of areas. Some of them are listed as following:

• Airports: Emergency response solutions are depend on context aware systems that can be automatically deliver and immediate security alerts to the relevant personnel at the airport, like : fire department, police, security, maintenance issues, etc. Services directly contacted to passengers, that are closely associated with the location of the passenger are developed as well. The notifications that are sent to the passengers’ mobile devices depending on the users’ location and inform them about, emergency exits way, shopping zones nearby, gates, arrivals and departures.

• Smart homes: Context awareness integrated with services and technology through home networking improves the quality of livelihood and can support disabled or senior citizen safety maintenance and independent lives in their homes. It can also provide Protective surroundings, Environmental information, Security functions, Appliances functions, etc.

• Leisure / Entertainment: This application area is closely connected with the user’s location. A variety of services have been lately developed that are offered and activated to the user and provide information about nearby concert events, maps, restaurants, theatres, shops, festivals and other information related to the area where the user is present.

• Hospitals / Healthcare: Context awareness can support to develop the quality of service in hospitals by implementing interactive hospitals where nurses and doctors are provided information about the diagnosis process, the treatment and medical history of patients in their proximity and they are able to carry out video conferences which enable collaborative feedback, diagnosis and discussion with other doctors. These interactive facility hospitals could also support to patients for intake medication, confirmation that the nurse is carrying the correct medicine, closely observe the patient’s current activity and raise emergency alerts if necessary, carry out video conferences between a doctor and a patient at home etc.

• Museums and fairs: Context awareness is closely communicated to detecting user’s position within a building and guiding them. It is commonly used to run applications on portable devices which are provided audio and video information and location sensitivity to the user about art pieces in a museum or a gallery, access directions about the path to reach a specific place of interest during a fair, send notifications about stands that might be of the user’s interest depending on his already known preferences and enable adaptation of sounds, music and other effects as the user changes the location.

• Offices: Environmental parameters measured within the office wherever a user is present and then inferring the activity of the user based on pending calendar entries etc. Beneficial work in the office is locating the position of employees in the company building. Additional information about the office where it is located (how many of them present and meeting in progress), the current status of the equipment in the company.

2.2 Challenges in Context Awareness

Due to the existence of a variety of different context-aware frameworks a series of challenges within the domain of context-aware computing occur. They deal with sensor issues, intelligence and inference, architectural design issues, privacy and ethical problems etc. These issues are clearly analyzed for more details as follows.

• Sensor issues:
  1. Frequently a component is unsuccessful afterward it’s manually restarted, which needs close human observation and maintenance of the whole architecture. To overcome this problem, redundancy process can be introduced as well as software components that confirm the normal operation of all critical entities in the system.
  2. To interact with the relevant actuators or the sensors is an important point of interest. Feedback the sensors and dynamically managing them during the operational phase can develop the quality and the applications performance of the service provided. Therefore, the need for improving a standard way to support this requirement.

• Intelligence and inference:
  1. A manner of autonomous learning without any user support requires to be added to the algorithms in use, so as to learn better from the new contexts that are presented to the system. In the sensor network, new contexts are continuously added for adaptive learning that has the issue draws more attention.
2. Moreover, context inference is a difficult task that needs a good mechanism for mapping simple captured context data to higher level ones. By using ontologies with logic reasoning, or probabilistic reasoning and it is difficult to complete. Research is still ongoing process to search the best technique that it will ignore the errors, automatically adapt the system to new types of data, reason correctly and learn autonomously.

Architectural design issues:

1. Designing approaches to make use of the context-aware system work with totally different levels of user collaboration is required within the systems. For example, self-adaptation of the system based on the user preferences should be provided, while the user preferences are not provided, the system should automatically adjust and continue working without that info. Moreover, adjusting the location where few calculations are performed (client or server side) depending on the info required to be evaluated is another architectural issue of interest.

2. Distributed and cooperative handling of context data should be added in the architectural designs developed. For example, a given device can be able to estimate some parameters that another device is not capable too. These descriptors are shared with other devices that require them, so as to enhance their knowledge base, should be modeled in the architecture design.

Other issues:

1. Privacy and security issues haven’t been quite solved because introduce its system were twofold. The amount of information which is on the application’s disposal increases, the application decisions’ quality automatically increases as well too. Applications manage more data and able to make more accurate decisions. Users have to be performed and manage the information that they are willing to reveal for allow their access and application purposes. Otherwise, the additional information is provided to the user and they offer less privacy. It becomes cumber some to constantly reduces its usability and reconfigure it. For that reason, as mentioned, needs better balance so as to preserve both advantages, privacy of users on one side, and confidentiality and trouble-free utilization on the other side.

2. Composition of simple context-aware services materializes like intuitive task and an obvious. It is a difficult task whose complexity lies within solving implementation issues and practical design that have not been predicted. In reality integration of the components of these systems is not seamless as it seems [18]. Hence, a better understanding on composition of context-aware services must be determined and guidance with best practices should be proposed.

III. CONCLUSION

The context-aware applications have to be assisted by means of adequate reasoning techniques and context information modeling for understanding that the development in the pervasive computing community. In context aware system, new opportunities entirely provided for end users and for application developers by performing and collecting with adapting systems and context data behavior consequently. By using these techniques to reduce the complexity of context-aware applications and improve their maintainability and resolvability of the context-aware systems. Particularly, these mechanisms of combination with mobile devices are utilized to increase usability and are of high value enormously. From this paper, clearly describe an overview of the context-aware computing in an effective manner and provides representation and inference of context and a comprehensive introduction to the definition. The significance of context aware modeling design indicates and challenges are faced during design process and system integration.

REFERENCES


