Survey on Context Aware Architecture

Miss. Pooja Kimatram Khatri  
Dept. Of Computer Science and Engineering  
Anuradha Engineering College  
Chikhli, Maharashtra.  
poojakkhatri23@gmail.com

Prof. Kishor H. Walse  
Dept. Of Computer Science and Engineering  
Anuradha Engineering College  
Chikhli, Maharashtra.  
kwalse1234@gmail.com

Abstract:-
The main characteristic of devices during a pervasive (or ubiquitous) ADP system is their context awareness that permits them to supply proactively tailored services to user and to applications consistent with the world context. So as to support the event and to ease the implementation of context-aware systems, several architectures were projected with characteristics associated with the applying domain and techniques used. A survey of such architectures that makes comparison between them and evaluates them is powerfully suggested. Projected surveys are either restricted to a restricted variety of architectures or don't provide an honest comparison or their analysis isn't supported applicable criteria that keep them as easy descriptions. Our aim is to create a survey of relevant architectures that mark the evolution of context-aware systems supported criteria associated with pervasive computing. This survey can function a guide to developers of context-aware systems and facilitate them to create subject selections.

Keywords: architecture, pervasive computing, context-awareness.

1. Introduction:-
Pervasive computing aims to supply proactively custom-made services to each user and applications per the world context. The most characteristic of devices in such system is their context awareness. Since its apparition, pervasive computing has needed tools (architectures, frameworks and middleware), ways and ideas to support the development of a context-aware system and ease their style and implementation. System design is made early within the development method and permits the creation of a high level style of the system that takes into consideration the fulfillment of requirements’ implementation. The design style is a very important step within the development of context-aware systems. Several researchers have projected many architectures, frameworks and middleware for context-aware systems with particularities associated with the appliance domain and techniques used. to guage these projected architectures, several surveys were done however they failed to cowl all architectures that mark the evolution of pervasive computing. They did not offer a solid comparison or evaluation and instead are simple descriptions. Even if they exist (in limited number of surveys), these surveys were not based on criteria related to pervasive computing particularities. Our aim, therefore, is to make a survey of relevant architectures that mark the evolution of context-aware systems beginning from localization-aware systems up to present context-aware systems. This survey presents a comparison and evaluation of architectures on various criteria which are considered important for pervasive computing such as: context abstraction level, communication model, reasoning system, extensibility and reusability. Our objective is to come up with a survey that will serve as a guide to developers and architecture designers of context-aware systems in a pervasive computing environment. The rest of this paper is organized as follows, in section 2. We review some previous surveys done until now on context-aware architectures and show their weaknesses

2. Related work:-
In the literature there have been many surveys on context-aware system architectures and for North American nation the foremost important ones ar the subsequent. Beldauf et al. [1] planned a survey of a decent range of architectures. They targeted particularly on superimposed architectures. Their survey was supported describing layers of various architectures and therefore the mechanisms utilized in every layer. In spite of the range of architectures cited, the survey wasn't done supported pervasive computing criteria and did not show clearly the strengths and weaknesses of every one. Kjaer [2] did a middleware minded survey however it deals conjointly with some subject aspects. It consists of a classification per a taxonomy judged vital by the author. This classification looks to North American nation a a lot of careful description of classical context-aware system design layers (sensor, interpretation, context management and adaptation) instead of the same survey. Because the former one, this survey wasn’t created so as to check architectures according to criteria connected to pervasive computing. Abshik associated Conway [3] represented four architectures however they failed
to create an analysis of them that renders their survey as an easy description. Wingrad [4] did a particular survey on structure models of architectures relative to context-aware human-computer interaction that isn't generic enough and unbroken his survey specific thereto domain. Finally Henrikson et al. [5] did a short survey as a vicinity of their work on 5 architectures that looks to North American nation the foremost attention-grabbing one. It makes a comparison of designs per criteria associated with pervasive computing however this survey doesn't cowl different architectures that mark the evolution of context-aware system architecture. The criteria used for comparison ar rather minded on distributed systems even supposing most of them ar basic for pervasive computing.

3. Evaluation criteria

A pervasive surroundings has some specific characteristics that ought to be taken under consideration once evaluating context-aware architectures. During this survey, we'll create associate degree analysis of those architectures supported some criteria that we tend to think about relevant for pervasive computing. These criteria are: a) level of context abstraction, b) communication model, c) reasoning system, d) extensibility and e) reusability. we've got chosen these criteria attributable to the subsequent reasons:

• Pervasive system uses sensors of various types to understand discourse data. Computer code design should hide the complexity of the physical devices by providing a higher level of abstraction that makes it freelance of physical sensor and enhances the reusability of design elements.

• Pervasive system consists of proactive devices that adapt to this context while not a precise intervention from the user. this needs that devices imbed a reasoning mechanism so as to require initiatives for an accurate adaptation.

• Devices should be autonomous, freelance from one another and might be simply connected. The peer-to-peer communication model appears the foremost acceptable for a pervasive system. It offers a straightforward thanks to tie devices and also the network are often discovered for a really modest investment and permits a straightforward sharing of discourse data among devices. It will want neither an ardent material (server) nor computer code (operating system, information base management system, etc.)

• A pervasive system is characterised by its speedily dynamic surroundings attributable to mobility; thus devices are often other or removed dynamically while not poignant the whole operation of the world system (hardware extensibility).

• Pervasive computing is a new domain of computing. Its design ought to give reusable elements so as to ease their integration and cut back development effort.

4. Context Aware Architecture

4.1 Active Bridge

The Active Badge project [6] developed by Olivetti analysis Ltd. aims to designed a system for phone calls delivery in line with the referred to as person’s localization. It permits the transport of telephone call to the phone nearest to the referred to as person. The system uses badges that incessantly emit infra-red signals at a given frequency. These badges area unit carried by personnel of associate enterprise and every badge contains the carrier identification. The signals emitted by these badges area unit perceived by some receivers distributed within the whole construction. The perceived signals area unit then sent to a server. The latter presents to a secretarial assistant the knowledge concerning badges carriers and their localization. This data helps the secretarial assistant to deliver a decision to the place nearest to the referred to as person (this task will be done automatically). The active badge is predicated on a distributed design of sensors. The superimposed design (figure 1) of Associate in Nursing application running on the server consists of the subsequent four layers:

- The network controller that supervises the operation of the detector network.
- The data presentation that is accountable for knowledge management and management of localization data.
- The knowledge process that selects the fascinating data at the time of localization variation.
- The user interface to show the matter data regarding badges variation position.

workstation in a very native space network. This prevents Parctabs from creating plenty of process that consumes their restricted resources. The communication system is predicated on the remote procedure decision (RPC) between ParcTabs and applications running on {a local|an space|a neighborhood} area network digital computer. This infrastructure permits the event of context-aware systems specifically those localization-aware, [for example|for instance|as Associate in Nursing example] we are able to cite Associate in Nursing incoming e-mail notification to a user supported his location and of close folks by displaying the e-mail text on the parcTab displayer or by a straightforward beep (filters are often accustomed advise users for under emergency cases once the user is attending an assembly or a conference). The authors contemplate several others context-aware applications supported this infrastructure like remote program management, aided collaboration, data and resources access per the context, etc. The ParcTab could be a primitive localization-aware system supported hardware infrastructure just like the Active Badge. The software system design is incredibly addicted to hardware and doesn't offer a decent abstraction of discourse data.

The active badge may be a hardware design for localization-aware system instead of a computer code design of context-aware system.
aware system with numerous computer code elements. It's specific to localization systems and can't be simply used for alternative reasonably context-aware systems. Finally it doesn't build any abstraction of discourse info (localization info during this case) that makes it terribly dependent to the hardware infrastructure.

4.2 ParcTab

The Xerox project ParcTab [7] may be a material infrastructure that enhances the event of applications aware to localization context (person location, encompassing devices, close individuals, etc.). The parcTab may be a personal digital assistant (PDA) carried by the user and operates as a graphical terminal. It uses infra-red communication with a transmitter Associate in Nursing exceedingly in a very area of Associate in Nursing building that communicates with space a neighborhood area network via an RS- 232 affiliation (figure 2). for every ParcTab there's a corresponding computer code agent that controls its communication with applications running on

![ParcTab infrastructure](image)

**Figure 1: The ParcTab infrastructure**

4.3 Stick-e-notes

The stick-e-notes project [8] may be a framework to support the development of context-aware application wherever localization is the basic element of context. During this system the most element may be a personal digital assistant (PDA) connected to a localization detector (GPS or Active Badge). The personal organizer could communicate with each other relying on the application. The plan behind the stick-e-notes comes from stick notes wont to prompt user regarding one thing (or to in brief describe something) and stuck on a door, a device, etc. during this case, the notes ar electronic and not hand written. Notes ar written by the user and ar hooked up to a particular context (example localization) and saved on his personal organizer. The electronic notes ar mechanically triggered (displayed by the PDA) whenever identical context seems within the future. For example: the user attaches an outline of a deposit once he visited one, anytime the user enter identical deposit, the outline note are displayed on his electronic device. Notes are also of various formats like text, HTML, sound, video, a program to execute, etc. The authors outlined four software package parts for the architecture:

- **SEPREPARE**: allows the user to organize notes.
- **SEMANAGE**: permits the management of notes
- **SETRIGGER**: allows notes triggering whenever similar context seems.
- **SESHOW**: allows the show of triggered notes and their storage Notes ar written in SGML language for simple data exchange. The stick-e-notes use a restricted set of discourse data (those associated with localization) ar hardware dependent (dedicated material) and don't offer a big improvement of context abstraction as compared with previous systems.

4.4 Cyberguide and Guide

The cyberguide project [9] equips user with a private electronic traveller guide tuned in to its context (localization, orientation, etc.). The hardware infrastructure consists of a collection of non-public digital assistants (PDA) connected to some world positioning systems (GPS) to observe a tourist's position. These PDAs will communicate in infra-red among them or with an {area|a neighborhood} area network. the target is to guide a traveller in his visit by providing him with fascinating sites to go to supported his actual location, methods to follow and some helpful data reckoning on his current position. The cyberguide design consists of the subsequent elements:

- Associate degree electronic geographical card of the physical surroundings visited by the traveller with a special illustration of exceptional objects (towers, park, museum, etc.)
- A browser that allows the detection of the traveller’s current location so as to supply him with data associated with the encompassing surroundings.
- A traveller that provides a message delivery service to the tourist to send request, suggestion, communication with alternative tourists and to receive broadcasted messages

Another project known as GUIDE [10] was planned with identical objectives because the cyberguide. For us, it looks that the 2 comes ar terribly similar with minor variations within the hardware used and internet access. These 2 comes ar specific to localization systems, they are doing not interpret discourse data (to come back up with a better level of abstraction), ar terribly dependent to the hardware used associate degree don't provide an protractile and reusable software package design.

4.5 CASS

The CASS tool [11] may be a middleware for supporting the event of context-aware applications. It provides an honest abstraction of discourse info ANd uses an object familiarized
model for context description. The design (figure 3) relies on a server containing a info of discourse info AnD a cognitive content with an reasoning engine to infer alternative discourse info employing a back chaining mechanism. The mobile devices square measure equipped with numerous sensors to understand context variation and send them to the server while not native process. Mobile devices and also the server communicate via wireless mode. The server additionally contains a module for context interpretation that gives it with the next level of abstraction. The design provides an honest modularity that enables straightforward modification of server parts above all the reasoning engine. The mobile devices don't build any process (all is finished by the server) that limits the autonomy required for pervasive systems however enhances the extensibility of the system (adding or removing devices need solely the configuration of the server). CASS additionally provides an honest abstraction of context thanks to its interpretation module and a reasoning mechanism that makes it additional proactive but the centralized design is its weakness (if the server is down all the system are going to be affected and becomes non operational).

A module for a hierarchical illustration of context so as to limit the particular scenario context then limit the set of attainable actions

Associate in Nursing illation engine that specifies the applications behavior to a given context and uses the execution model event-condition-action The communication between sentient objects, sensors and actuators that compose the system uses the mechanism primarily based on events that square measure established dynamically throughout the system operation. This design presents several blessings as earlier declared however remains an advertisement hoc answer for a mobile network. The illation engine written in CLIPS language needs qualified personal to make, modify or adapt it to Associate in Nursing alternative application that limit its usability. The invention mechanism isn't well elaborated by authors and doesn't permit a measure of the extensibility of the design. Also, the model of context used doesn't offer a whole set of discourse info required for adaptation task.

4.7 Context management framework

The CMF (context management framework) [13] permits linguistics reasoning on context in real time and even within the presence of noise, doubt and speedy variation of context. It delivers discourse info to applications by exploitation a communication model supported events. The framework proposes a client/server (figure 5) design composed of the subsequent basic components:

- Context manager: accountable for the storage of discourse info on server and also the delivery of context to shoppers exploitation completely different sorts of mechanisms (request/response, subscription/notification, etc.)
- Resource server: accountable for the acquisition of discourse info from physical sensors and their interpretation according to a specific format before causation them to the context manager
- Context recognition service: to blame for the conversion of [the data|the info|the info] stream to a presentation outlined within the context metaphysics

- Amendment detection service: accountable for the detection of service amendment and thus the context amendment
- Security: accountable for the verification and management of discourse information The CMF uses metaphysics for context illustration however doesn’t provide a context reasoning module. It contains a decent mechanism for context interpretation that provides a decent abstraction of context and enhances the reusability additionally to a module for context security. It uses a server for context management (centralized system) that is that the main downside since, once the server is down all the system are going to be affected and renders the devices less autonomous that are some things not fascinating in a very pervasive ADP system.

4.8 JCAF

Access Aware Awareness framework supported java programming language to support the event of context- aware
applications. The JCAF design consists of a group of elements referred to as “context service” communication in an exceedingly peer-to-peer mode. These elements square measure to blame for assembling context info in an exceedingly specific surroundings (room, hospital, laboratory, etc.). A context service contains four modules as follows:

**Figure 3: Context Service Modules.**

- **Entity container:** accountable for context exchange with context purchasers by employing a communication mechanism supported events (subscription/notification). It conjointly contains one or additional entities that describe the context of Associate in Nursing surroundings object (person, computer, doctor, patient, etc.).
- **Surroundings entity:** permits communication between entities and management access to shared resources.
- **Access control:** controls access to the entity via correct authentication of client’s question to access entity context.
- **Entity listener:** it will be an entity of another context service and might access the entity context of a context service either by the request/response theme or the subscription/notification theme. It’s attainable to use the subscription/notification theme in keeping with the kind of context.
- **Context monitor:** permits the acquisition of context via sensing elements and makes transformation of crude context.
- **Context actuator:** permits commanding the actuators of the physical surroundings. The JCAF conjointly control the discourse info (trust on the data detected by a specific sensor, error chance of knowledge perceived by a sensing element, etc.). The remote communication between the design elements is completed victimisation java RMI (remote methodology invocation). The context service doesn’t have Associate in Nursing automatic discovery mechanism however will use a configuration file containing all others active context services. The JCAF doesn’t have a context reasoning mechanism and doesn’t offer a decent abstraction of context as a result of there’s no part that produces context interpretation in a definite manner. the dearth of Associate in Nursing automatic discovery mechanism limits its extensibility however the JCAF offers reusable and moveable modules due to its use of java language.

4.9 Context toolkit

The context toolkit [15] was planned as a tool to assist the developers of context-aware systems. It’s a bedded design that allows the separation of context acquisition, illustration and adaptation method. It’s supported context widgets that operate equally to graphical program widgets so as to cover the complexity of physical sensors. These widgets supply a decent abstraction of context and supply reusable blocs for context sensing. The design (figure 7) consists of the subsequent elements: • **Sensor:** sensing of physical context. • Widgets: change the encapsulation of discourse info offer[and supply) ways to access them within the same manner as graphical widgets • **Interpreters:** build context transformation so as to provide a higher level of abstraction of context • **Aggregator:** makes context grouping per a topic or a scenario.

- **Discoverer:** maintains a register of existing capabilities within the framework (presently accessible components to be used by applications).
- **Service:** executes actions for applications This design is simple to implement, offers a distributed communication among system devices and reusable widgets however the invention mechanism is centralized that doesn’t build it an ideal peer-to-peer communication model. It has a restricted extensibility once the amount of devices will increase. The design takes into consideration events (to appraise context variation) by employing a thread for every event that overloads the system and affects its performance. The design will not contain a layer or a module for context reasoning as a result of the model used for context illustration (key/value) doesn’t allow a decent reasoning.

4.10 Atomic numbers 1

Hydrogen [16] is associate degree design and a framework for context-aware systems. It’s a 3 superimposed design that responds to explicit necessities of mobile devices. The design has the subsequent layers: adaptation, management and application. The context server (management layer) contains all the detected info perceived by the sensors of the adapter layer and provides context to the applying layer of the hooked up device or alternative devices employing a peer-to-peer communication model. The atomic number 1 approach considers context as associate degree pertinent info on associate degree application surroundings and describes it mistreatment an object headed model. The design will be enforced simply, is easy and takes into consideration the restricted resources of mobile devices (battery, memory, processing, etc.) and uses a peer-to-peer communication model (distributed). The adapter layers will each the sensing and also the interpretation task of context that doesn’t provide
an honest abstraction of context and limits the reusability of such part. Also, it makes it terribly dependent to sensors. The design doesn't contain a reasoning module on context to ease the variation task.

4.11 SOCAM

SOCAM [17] is associate degree design of a service headed context-aware middleware for building associate degree speed prototyping of context-aware mobile services in an intelligent automotive. The design consists of the subsequent components: context supplier, context interpreter, (context information and context reasoner), service locating service, context- aware mobile service and context information. The design uses the client/server model wherever the context interpreter collects discourse info from context suppliers (internal or external) and context information and provides them to the context- aware mobile services and also the service locating service. The most strength of the SOCAM design is its context thinker that uses metaphysics for context description and permits a strong reasoning on context. It uses 2 categories of ontologies: domain specific and generalized ontologies. many reasoning systems will be incorporated within the context interpreter to support a range of reasoning tasks. The design was projected to support the event of a little non distributed application (intelligent car) that limits its use during a wide selection of pervasive computing applications. The context interpreter is full with a very important quality of data (ontologies of various domains) that affects the world performance of the system however enhances its reusability, additionally to the most important downside of a centralized design that contradicts the nature of a pervasive system that is a distributed one with autonomous devices.

4.12 CoBrA

CoBrA [18] is a design supported broke Agent to support the event of context-aware applications in AN intelligent house. The broker is AN autonomous agent that manages and controls the context model of a selected domain. It runs on a fervent laptop (server) with powerful resources. The broker agent encompasses a bedded design containing the subsequent components: context information, context ratiocinator engine, context acquisition module and privacy management module. The broker agent collects context from devices, alternative agents and sensors of its close surroundings and makes their fusion in a very coherent model which is able to be shared among devices and their corresponding agents. elapid uses metaphysics for context description that permits a decent reasoning and a far better sharing of discourse data. It uses a centralized model for the storage and also the process of context so as to avoid wasting the restricted resources of mobile devices and uses a confidentiality policy for the user. The design needs a fervent server for the broker that will increase its price and limits its usability additionally to the matter of a centralized design.

- Reasoning: (not gift all told architecture) deduces and predicts new discourse data.
- Storage and management: basic operation in managing discourse data (ads, remove, research, update, etc.)
- Adaptation: adaptation of provided services consistent with this context. The projected architectures are principally specific to AN application domain (localization systems, human- laptop interaction, etc.) and need further effort for his or her adaptation to alternative domains.

Architectures supported a server suffer from the matter attributed to a centralized system: once the server breaks down, all alternative system parts are affected additionally it needs a fervent hardware and package that will increase its implementation price. A centralized design contradicts the character of discourse data in a very pervasive ADPS that is in general distributed and the quality characteristic of devices in such surroundings. Rare ar the architectures that contain all the layers mentioned on top of and most of them don't use a sound and reliable context model which allows economical reasoning and eases the difference task. Context modeling is out of the scope of this paper however it's key idea for design style (context management layer and reasoning layer), A survey created by Strang et al. [19] Containing a fascinating comparative study of various modeling strategies concludes that metaphysics SOAP + RDF/OWL FIPA-ACL + RDF/OWL makes the most effective description of context compared to alternative strategies. A pervasive system is characterized by its apace dynamic surroundings thanks to mobility; therefore devices is extra or removed dynamically while not moving the whole operation of the world system which needs a dynamic and automatic devices and resources discovery mechanism. This Figure 10: The elapid design
5. DISCUSSIONS

Except for primitive localization-aware system design, most of the planned architectures build distinction between context sensing processes from its use. This allows Associate in Nursing abstraction of low level sensing details and will increase the extensibility and reusability of design parts. Among planned design, there are 2 approaches betting on whether or not discourse data are centralized or distributed. Most of those architectures are bedded and composed of the subsequent components: • Sensor: physical sensing of discourse data. • Interpretation: transformation of crude data into a additional vital and helpful data. Aspect wasn’t deeply mentioned in most architectures and desires additional attention in future systems. Subject field style of context aware systems wants additional efforts so as to supply Associate in nursing applicable design that suits pervasive system necessities. The table below summarizes characteristics of surveyed architectures (table1).

### Table 1: Characteristics of Architecture.

<table>
<thead>
<tr>
<th>Activity</th>
<th>Intent</th>
<th>Abstract Step</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parking</td>
<td>Make sure the parking proceeds smoothly</td>
<td>Arrange the parking place</td>
</tr>
<tr>
<td>Belling water</td>
<td>Prepare for making tea</td>
<td>Fill teapot with water</td>
</tr>
<tr>
<td>Pouring boiled water into a teapot</td>
<td>Control temperature of boiled water</td>
<td>Wait for the boiled water to cool down</td>
</tr>
<tr>
<td>Warning the teapot</td>
<td>Scald /leap from hygienic reasons</td>
<td>Pour hot water from the teapot into the teapot</td>
</tr>
<tr>
<td>Seating tea with a teapot</td>
<td>Optimize tea flavor</td>
<td>Scoop up proper amount of tea with tea-leaves</td>
</tr>
<tr>
<td>Seating tea</td>
<td>Remove impurities from among tea leaves</td>
<td>Pour hot water from the teapot into the teapot</td>
</tr>
<tr>
<td>Serving</td>
<td>Control the strength of the tea</td>
<td>Pour hot water from the teapot into the teapot</td>
</tr>
<tr>
<td>Pluton</td>
<td>Make sure each cup of tea is of the same taste quality</td>
<td>Pour hot water from the teapot into each/fragrance cup in order</td>
</tr>
<tr>
<td>Initial tasting</td>
<td>Assess the taste</td>
<td>Small and taste a little (Pour tea from the fragrance cup into a sipping cup in order to do so)</td>
</tr>
<tr>
<td>Smelling and drinking</td>
<td>Expect served tea</td>
<td>Small and taste a little (Pour tea from the fragrance cup into a sipping cup)</td>
</tr>
<tr>
<td>Drink tea</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
for future systems. This survey aim is to serve as a guide to offer a useful recommendation to developers and designers of context-aware systems and help them decide on available architectural choice.

REFERENCES


