# Mobile Database in Pervasive Computing

**Prof. Smita Krishna Patil[1] Prof. Samira Francis Nigrel[2] Prof. Rasika Patil[3] Prof. Shweta Sharma[4] Prof. Chandana Nighut[5]**
[1,2,3,4,5]Assistant Professor
[1,2,3]Department of Information Technology [4,5]Department of Computer Engineering
[1,2,3,4,5]Atharva College of Engineering University of Mumbai, India

*Abstract*— This paper proposes a mobile database in pervasive computing system. This system is useful for finding an open parking space in an enormous parking area. Graphical images of parking area are stored in the mobile database. When car enters in the parking area, pervasive system automatically notifies the driver about the available parking space. First, pervasive system sends a query to the database and gets the reply and notifies driver about the available space. With the help of service discovery and service location protocol the system is implemented. These protocols are used to send and execute queries with MoGATU framework. Mobile database is managed by data management system.

***Key words:*** Data Management, Mobile Database, Pervasive Computing, Query Processing

## I. INTRODUCTION

Computing is no longer a isolated activity bound to a desktop; network computing and mobile computing [2,4,7,8] are fast becoming a part of everyday life and so is the Internet. Rather than being an infrastructure for computers and their users alone, it is now an communications for one and all. We expect devices like *PDAs* (Personal Digital Assistants), mobile phones, offices PCs and even home entertainment systems to access information and work together in one integrated system and the dare is to merge these technologies into a seamless whole and on the Internet. The aim of Pervasive Computing [1,5,6] is for computing available wherever it's needed. It spreads intelligence and connectivity to more or less everything. So conceptually, ships, aircrafts, cars, bridges, tunnels, machines, refrigerators, door handles, lighting fixtures, shoes, hats, packaging clothing, tools, appliances, homes and even things like our coffee mugs and even the human body and will fixed with chips to connect to an infinite network of other devices and to create an environment where the connectivity of devices is embedded in such a way that it is unobtrusive and always available. Insidious computing, for that reason, refers to the emerging trend toward numerous, easily accessible computing devices connected to an increasingly ubiquitous network infrastructure [9].

Recent advances in portable and wireless technology have led to mobile computing[4,8], a new dimension in data communication and processing. Portable computing [5] devices coupled with wireless exchanges allow clients to access data from virtually anywhere and anytime. This feature is especially useful to geographically dispersed organization. It include electronic valets, news reporting, brokerage services and automated sales forces. A mobile database [3,7,8] is a record that resides on a mobile device such as a PDA, a smart phone, or a laptop. Such devices are often restricted in resources such as memory, computing power, and battery power. A mobile database is managed by a Database Management System. All over again, due to supply constraints, such a system often has limited functionality compared to a bursting blown database management system. For example, mobile databases are lone user systems, and therefore a concurrency manage mechanism is not required. Other DBMS modules such as query processing and recovery may also be imperfect.

Queries to the mobile database are habitually posed by the consumer of the mobile device [6]. Updates of the database may initiate from the user, or from a central server, or directly from other mobile devices. Revises from the server are communicated wirelessly. Such message takes place either via a point-to-point connection linking the mobile device (the client) and the server, or via broadcasting by the server. Through updates from extra mobile devices may use dumpy-range wireless statement protocols such as Bluetooth or Wifi.

Invasive computing intends to make our lives trouble-free through the use of tools that allow us to handle in turn easily. These "tools" are a new class of intelligent, moveable devices that allow the user to plug into potent networks and gain through, simple, and secure access to both related information and services. Pervasive computing devices are not own computers as we learn to think of them, but incredibly small - even invisible - devices, whichever mobile or embedded in almost several type of object conceivable; all communicating during increasingly interconnected networks. In order instantly available anywhere and anytime is Pervasive Computing.

## II. PERVASIVE COMPUTING CHALLENGES

This section describes the pervasive computing challenges with different scenarios.

### *A. Challenges*

There are few challenges [1] in pervasive computing, so we will try to find the solution with the help of some scenarios.

- Invisibility – desertion of computing technology from users' consciousness; embedding/combining computing infrastructure with building infrastructure
- Scalability – scalability of users, machines, bulk of interactions in a local area

- Availability – access to software applications and information anytime and anyplace
- Dynamic – users and devices are portable, services are offered by collaborating distributed components
- Heterogeneity – variety of hardware, software platforms, network protocols, service providers
- Integration with people – personal privacy, user intentions, access control

### *B. Example 1*

John opens his refrigerator to obtain out a soda, and he notices that there is only one can left. John then scans the UPC of the soda with the scanner attached to the refrigerator. John plans to have guests over this weekend, and makes a note on the refrigerator device that he needs to replenish his soda by Friday. The next day, on his way home from work, John happen to approach a local supermarket. His car signals to John that he is near a grocery store, and if it is convenient, that he should stop by. Suppose John does not act on the opportunity, Friday rolls around and he still have not bought the drinks, a notification will be sent to John's PDA by the refrigerator to alert John.

John's virtual shopping list is maintained by his refrigerator. When John scanned the soda, the refrigerator adds soda to the shopping list. When John notes down Friday, the refrigerator notifies his car about the need for soda. When John is on the road, his car constantly monitors the available services; when his car detects the supermarket service, it notifies John. When it's Friday, and the refrigerator has not received a "soda purchased" notification, it sends a notification to John's PDA.

### *C. Scenario 2*

It is 5:40 in the afternoon and Bob's work day is just ending. As he is getting ready to leave the office his phone rings. It is Alice asking him to meet her at the local shopping mall. Bob agrees to meet her and notifies his mobile device about the appointment. While he is walking through the building toward the parking lot, his device is able to connect to the office network infrastructure using a WLAN and fetch the directions through an information broker. In the car, the mobile device sends directions to the on-board computer that displays them on the area map. While on-route, Bob feels that the traffic is not moving fast enough and would like to get to his destination via some quicker route. He instructs his device, which can now connect to cars around him, to ask these if they know of a faster route. The device contacts its neighbors and returns with an alternate map. Although the suggested route is longer it circumvents afternoon traffic jam building up on the original route. Bob takes the different roads and arrives at the mall's entrance twenty minutes earlier.

He decides to use the extra time by checking out local stores for a bargain on some small gift for Alice. His mobile device finds out from the mall directory server about stores that sell such gifts, contacts them to find out possible gifts within a $25 and presents ideas to Bob. Similarly, the mobile devices have a wish list of other garments Bob wishes to purchase and proactively search for these bargains as well. Upon Alice's arrival, Bob asks his mobile device to suggest available restaurants (the device cached such information during the mall exploration) and lets Alice pick one. She chooses the closest Italian restaurant that indicates it has an available table with no waiting period. Thus, they are seated immediately and spend several hours eating and chatting.

## III. RELATED WORK

In this paper, we propose a service directory and data management [1,2] approach. Goal of service directory [6,9] is to allow devices to advertise, discover, configure, and communicate with each other in a mobile network. Here we will use mobile database for storing different locations, data and contacts which will be useful for communication.

When a device is moving between networks, it needs to know what services are still available, now available, and no longer available. It also needs to know how to contact these services. Then, based on the resources, the device would able to perform its task. For this we will use two protocols, service discovery protocol and service location protocol.

Service Discovery Protocol [1] – protocol to facilitate association of devices that have services to offer or that need services. It enables interoperability between devices that can communicate with each other. Also accommodate heterogeneity, devices can be of different kind and minimize administrative overhead, increase usability. A service discovery protocol for advertisement and discovery of network services. Each service has a service type, which defines a collection of services with a common nature. Each service has a name and a list of characteristics. Each characteristic is defined by a attribute-value pairing. Service with the same service type have similar attributes but different values.

### *A. MoGATU Framework*

The past few years have seen significant work in mobile data management, typically based on the client/proxy/server model. Mobile/wireless devices are treated as clients that are data consumers only, while data sources are on servers that typically reside on the wired network. With the advent of pervasive computing environments, an alternative scenario arises where mobile devices gather and exchange data from not just wired sources, but also from their ethereal environment and one another.

This is accomplished using ad-hoc connectivity engendered by Bluetooth like systems. The traditional client/server model [9] is no longer appropriate to represent the new scenario; instead, mobile devices should be treated as peers. The mobile devices become both data consumers and providers, spontaneously interacting with other devices in their vicinity to pursue their individual and collective goals. The data-intensive, pervasive computing model for mobile ad-hoc networks poses new challenges for data management among these devices. These challenges include the spatio-temporal variation of data and data source availability, lack of a global catalog & schema, no guarantee of reconnection among peers, no guarantee of collaboration among peers, and the issues of commits and aborts due to the serendipitous nature of the environment.

In this paper, we are solving the problem of data management in pervasive computing environments facing the above challenges. Our solution broadly consists of two parts: treating the devices as semi autonomous entities guided in their interactions by profiles & context, and the design of a contract-based transaction model [3,4,7]. The profile is grounded in a semantically rich language for representing information about users, devices, and data objects each described in terms of "beliefs", "desires", and "intentions" - a model which has been explored in multi-agent interactions. We introduce data-based routing algorithms and semantic-based data caching and replication algorithms enabling mobile devices to utilize their data-intensive vicinities. We also introduce novel interactions autonomously initiated by each device while trying to pursue its user's "intentions" and "desires" encoded in the user profiles. We are developing transaction models based on contract net principles for peer-to-peer interaction and will demonstrate our findings and solutions by creating a lightweight peer-to-peer data management architecture in pervasive environments. We also investigate cross layer security issues, integrating data level trust with network level observations.

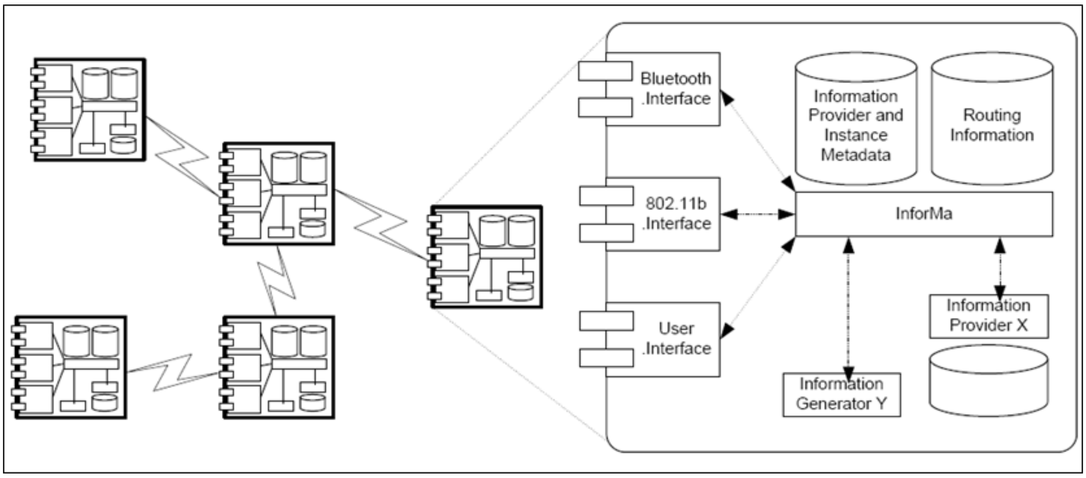


Fig. 1: Entity details and interactions in the MoGATU framework

## IV. PROPOSED WORK

Today's "inch-scale machines" come in many forms, from the active badges that, to Smartphone's, which could be considered the direct descendants of Weiser's tabs. But today's inch-scale machines go beyond badges and tabs to include devices such as wristwatch computers, sensor motes, MP3 players, and handheld games. Inch-scale computers are almost ubiquitous, but compared to Weiser's vision, most are used less as badges or for sensing or controlling our environment and more for personal productivity, entertainment, and communication.

In this paper we will create a system which will work with mobile databases in pervasive computing environment, where we will try to find an open space for car parking. For this we will use service discovery protocol for finding an open space. All the details related to the parking area, already stored in mobile database. When we will enter in parking area our pervasive system will that is Smartphone or PDA will automatically notifies that we are currently in parking lot. And pervasive system will work as follows.

### *A. Query Processing*

Monitoring physical-world phenomena (*events*) of interest, pervasive computing applications integrate data acquisition, communication and operations (*actions*) on small devices that are embedded or mobile in the world. For instance, in our scenario, a surveillance application automatically operates remotely-controllable cameras to take photos of open parking space based on the available data in mobile database. In the meanwhile, it sends the photos to the cell phone or PDA of the driver who is currently driving the car. Due to the heterogeneous and resource-constrained nature of the devices involved, to date pervasive computing applications are usually difficult to develop and optimize, and their functionality and usefulness are limited.

First pervasive computing applications allow specifying actions on devices through database queries [6]. Then uniform data communication layer that manages a number of networked, heterogeneous devices. This layer ensures that the system, not the individual applications, is responsible for monitoring and tuning the current network infrastructure and the physical status of the devices. Finally, there is an action oriented query processing engine for queries embedded with actions in the middle. Being the core of our framework, this engine is responsible for generating, optimizing and executing Query plans.

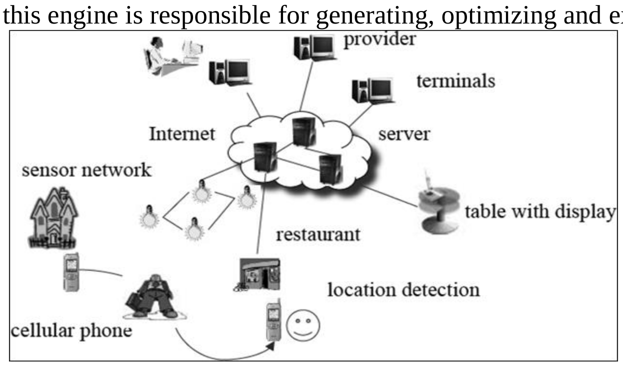


Fig. 2: Basic structure of pervasive computing

As shown in the fig. 2 with the help of service discovery protocol system will find a efficient service and using service location protocol it will get the images of open space and notifies driver.

### *B. Modeling a Pervasive Environment*

In order to homogeneously represent data sources and other resources from pervasive environments [1,5], we propose a model that integrates distributed functionalities of resources within data sources. Our model, based on the relational model, is built on the following notions: prototypes, services and extended relations with virtual attributes and binding patterns. Distributed functionalities can be represented as services implementing prototypes. For example, a surveillance camera take photos of open space from the environment that implement a prototype takePhoto():(photo) that takes zero input attribute and provides one output attribute photo; a mail server, an instant messaging server and a SMS gateway are three services that implement a prototype sendMessage(address,text):(sent) that takes two input attributes address and text and provides one output attribute sent.

### *C. Designing a Pervasive Environment Management System*

In order to validate our approach and conduct some experiments, we have designed and developed a prototype of a Pervasive Environment Management System (PEMS) [5]. The role of a PEMS is to manage a relational pervasive environment, with its dynamic data sources and set of services, and to execute continuous queries over this environment. The PEMS architecture is composed of three core modules (Environment Resource Manager, Extended Table Manager, Query Processor) and several distributed modules (Local Environment Resource Managers): the deployment of the different modules and their interactions are illustrated in Fig. 3.

The core Environment Resource Manager handles network issues for service discovery and remote invocation, as well as input of data from remote sources (data relations, data streams). It discovers and communicates with Local Environment Resource Managers that are distributed in the network. Services simply register to their Local Environment Resource Manager, and are then transparently available through the core Environment Resource Manager. The Extended Table Manager allows to define relations from Serena DDL statements, and to manage their data (insertion and deletion of tuples). The Query Processor allows to register queries using the SQL and to execute them in a real-time fashion. It implements all relational operators and realization operators, as well as the Window and Streaming operators for continuous queries. Service invocations are handled asynchronously by the invocation operator, relying on the core Environment Resource Manager for actual invocations. The Query Processor also handles service discovery queries: it continuously updates some specific relations so that they represent the set of services (implementing some given prototypes) that are available through the core Environment Resource Manager, like for the cameras.

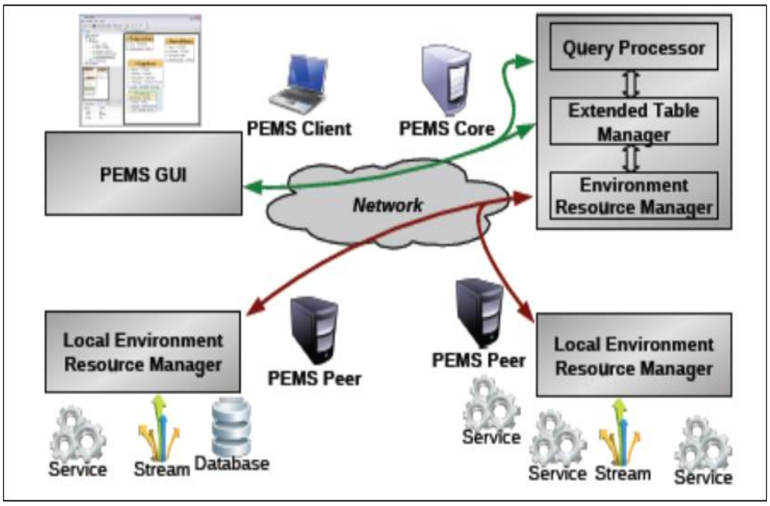


Fig. 3: Overview of the PEMS Architecture

## V. CONCLUSION

Pervasive computing provides an attractive vision for the future of computing. Well, we no longer will be sitting down in front of a PC to get access to information. In this wireless world we will have instant access to the information and services that we will want to access with devices, such as Smartphone's, PDAs, set-top boxes, embedded intelligence in your automobile and others, all linked to the network, allowing us to connect anytime, anywhere seamlessly, and very importantly, transparently. Computational power will be available everywhere through mobile and stationary devices that will dynamically connect and coordinate to smoothly help users in accomplishing their tasks.

## REFERENCES

[1] http://ewh.ieee.org/r10/bombay/news4/Pervasive_Computing
[2] Cache Management for Mobile Databases: Design and Evaluation Boris Chan Si Hong. Lcorig Department of Computing Polytechnic University Hong Kong 1995 IEEE

[3] A Distributed Mobile Database Architecture Anne Marie AMJA, Abdel OBAID1, Normand SEGUIN 2011 IEEE Asia -Pacific Services Computing Conference
[4] An Approach of Mobile Database Design Methodology for Mobile Software by SolutionsWeider D. Yu, Tamseela Amjad, Himani Goel, Tanakom Talawat The 3rd International Conference on Grid and Pervasive Computing – Workshops 2008 IEEE
[5] Pervasive Tabs, Pads, and Boards: Are We There Yet? PERVASIVE computing Published by the IEEE CS n 1536-1268/12/$31.00 © 2012 IEEE
[6] Systems Support for Pervasive Query Processing by Wenwei Xue Qiong Luo Lionel M. Ni Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICSCS'05) 2005 IEEE
[7] A Survey of Mobile Database Security Threats and Solutions for It by Parviz Ghorbanzadeh Aytak shaddeli Roghieh Malekzadeh Zoleikha Jahanbakhsh
[8] A Mobile Database Design Methodology for Mobile Software Solutions by Weider D. Yu Sunita Sharma 31st Annual International Computer Software and Applications Conference (COMPSAC 2007) 0-7695-2870-8/07 2007 IEEE
[9] A Three-tier Architecture for Ubiquitous Data Access by Sumi Helal, Joachim Hammer, Jinsuo Zhang, and Abhinav Khushraj