SapienNet: Dissolving Communication Boundaries

Application ID: 66BJGW

Region ID: 10

Institution:
Department of Computer Science
Indian Institute of Technology, Guwahati
North Guwahati, Assam
781039 INDIA

Student Mentor: Dr. Shivashankar B. Nair (sbnair@iitg.ernet.in)

Students:

Gautam Das (gautam@iitg.ernet.in)
Rahul Singh (rsingh@iitg.ernet.in)
Suvesh Malhotra (suvesh@iitg.ernet.in)
Archit Gupta (archit@iitg.ernet.in)

Submission Date: 23rd Apr. 2005

GOING BEYOND THE BOUNDARIES

1. INTRODUCTION

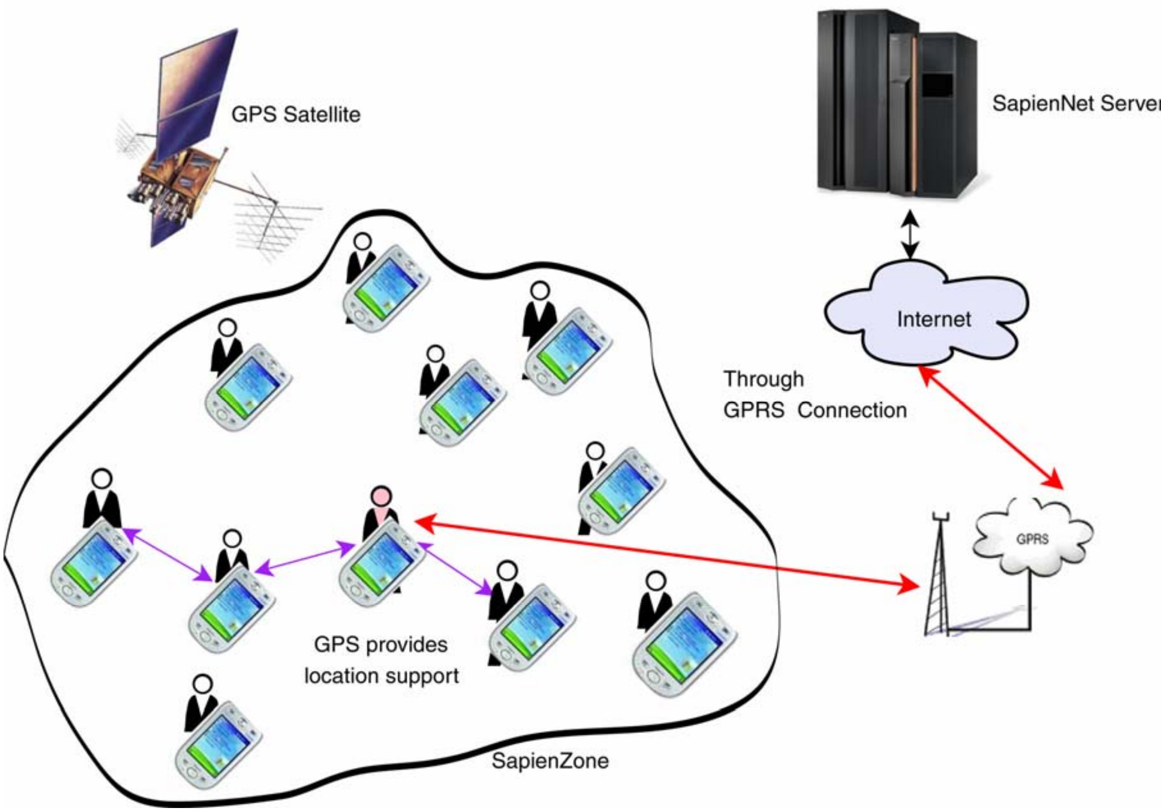
The advent of ubiquitous and pervasive computing facilitated by the Internet has opened up a plethora of applications. It is now possible to see and communicate with people around the globe. Nevertheless, there is still much more to be done to knit a closer social fabric. Despite its popularity and immense potential, the Internet has still not been fully exploited to satisfy true social human-human interactions. Locating a person whom one has little idea about in a nearby crowded place, still calls for an extroverted attitude. For instance our own experience of finding an English speaker in a non-English speaking country in spite of all the facilities offered by the modern digital world has been frustrating. We would greet every potential candidate on the streets and query to find whether they knew this tongue. Even with an assortment of communication technologies, the ability to know about people in one's vicinity seems to be grossly lacking. Take for instance the predicament of an unknown person who is sent to pick you up from an airport. The best but embarrassing alternative for the respectable gent would be to hold a placard with your name on it and wait at the exit. Another instance is at an International conference where you would like to know more about the current speaker or the delegate beside you before venturing for a personal discussion. While these may look like trivial situations, critical ones where it is imperative to find the right person as quickly as possible also call for such a facility. A patient in need of immediate attention has been advised to consult Dr. John Smith whom he is never met before. The Doctor's whereabouts, apart from the fact that he is around someplace in one of the crowded wards of the hospital, is not known. A mobile phone will not be able to trace him without disturbing his current activity. Furthermore what he looks like still remains unknown to the patient.
To overcome this, we propose a *location-aware* proximity network to break the fetters of the digital-only world and pave the way for better and unobtrusive social communication. Christened ***SapienNet***, our system facilitates free and unrestrained communication amongst individuals and promotes the formation and knitting of human communities. This network coaxes human-human interactions by allowing searches to be carried out on a variety of information about people in close proximity. It serves impromptu information of such people and also sets an alert for the presence of friends around the user. ***SapienNet*** is thus as *an attempt to bring meaningful information about the people in the vicinity of a user.* It encourages even introvert humans to know about other persons thereby breaking their inherent diffidence and bringing them closer, socially. Thus without direct interaction with the people around, it allow users to unperturbedly wean information on the likely persons he/she wishes to interact.

SapienNet will not only facilitate finding whether your friend is there in the crowded Times Square in New York City where an estimated 26 million tourists flock from around the world, but will also help identify the profession of the shy person sitting beside you. The latter case becomes significant especially in countries that have a more conservative culture and where impromptu social interactions remain inhibited for a wide variety of reasons. Embedded in a handheld device or in a mobile phone, ***SapienNet*** has the capability to act as catalyst in transforming introvert beings into outgoing ones thus breaking social boundaries without upsetting or invading human sentiments.

SapienNet allows for ***PeopleSearch*** a facility to search, identify and locate desired persons in a non-intrusive manner. It also features ***EarlyIntro*****,** a component that empowers diffident users to explore and search for more information on specific people in his/her vicinity before entering into a verbal conversation with them. An ***ActiveAlert*** feature aids in identifying as and when the desired person comes into his/her vicinity.

2. SYSTEM OVERVIEW

The users of SapienNet form a spontaneous, location-aware, wireless mobile ad-hoc network. Each user possesses a GPS equipped PDA with integrated 2-way wireless capabilities (GSM/GPRS and WLAN - 802.11) and SapienNet system software installed in it. The wireless interface of the PDA is used to form the ad-hoc network. The GPRS is used to connect to the SapienNet web services, which are available through the Internet. This SapienNet software enables the user to get the required information about the people in his/her vicinity. Vicinity in SapienNet is referred to as the SapienZone. The SapienNet server hosts a priory information of all users. This information is made available to other users through web services. The user can search, identify and locate a desired person in his/her SapienZone. He/she can also get introduced to new people, obtain a live map with user locations and even bookmark people of interest among other things.



2.1 Performance Requirements

The SapienNet architecture should have the following qualities:

- **Scalability:** Since handheld devices are intended to be pervasive, the architecture should be able to handle a large number of users.
- **Robustness:** The architecture should be able to handle a varying density of users as also sudden spikes in the same. The web services should be able to a serve large number of requests.
- **Low Network Latency:** The information about the presence and location of a fellow user should not be stale. While real-time guarantees cannot be given in an ad-hoc network, network information must be refreshed at a reasonable rate.
- **User's Privacy and Authentication:** The user must be able to set access levels to the *a priori* information supplied. A malicious user should not be able to assume a user's identity.

2.3 Design Methodology and Team Organization

We followed the **Extreme Programming (XP)** [6, 15] software model. Since, the team had prior experience in applying the Rational Unified Process (RUP), useful ideas of the RUP were incorporated to formalize some of the activities of XP that demanded strict discipline. This formalization helped in keeping the team on track and did not let the lack of experience in XP affect team performance. We chose XP since the time available from conceptualization to finished product was limited and constrained by other academic commitments during the semester. We wanted parsimony in the development process so as to remove unnecessary activities. Since we are a small, well-knit team, we wanted a model that had a collaborative and communicative working style so as to achieve maximum efficiency. The design of the system was object oriented and software components were developed modularly.
The team consisted of 4 members, of which one was the **Team Leader** with the overall responsibility to keep track of progress of the project as planned. The team was divided into two pair-programming sub-teams as per XP guidelines. One sub-team deployed and tested the web services and databases on the server side. On the client side (PDA) they developed software components which consumed the web service, designed and developed the user interfaces. The other sub-team developed a new message routing algorithm for the ad-hoc network. They used the algorithm to develop the network software component. They also worked on hardware interfacing of the embedded device with the GPS, GPRS and the external ROM.

2.4 Innovation

Most of the handheld devices currently available (including PDAs, mobile phones, etc.) have facilities to communicate with people. To find out more of a person one has to essentially access the web and then wean out information on a desired person. Much depends on which search engine we use and whether the information is actually available. An English speaking Indian in a remote South Korean town wanting to find his way back to his hotel would find the task extremely difficult. In many a country initially start-up help available on mobile phones too are provided in the national language. Where applications running on mobile hand held devices fail to come to the aid, ***SapienNet*** can easily come to the rescue of people in alien lands by bringing together the like-minded. It also allows users to click on points, on a live map on the device, that indicate people in the vicinity, to get complete information about the person (eg. Profession, address, interests, photo, age, etc.) positioned relatively in the real world. This has a semblance of virtually saying *Hi* to your neighbor and getting acquainted with him/her. This emphasizes how the system attempts to break its digital boundaries to create a closer social network in the user's proximity. This innovative nature of ***SapienNet*** could be attributed to the much-used adage - *Birds of a feather flock together*.

2.5 Social Benefits

A constructive and intelligent society can be realized only when each entity does his/her part to the best. To achieve this all the necessary help should be rendered to every individual at appropriate times to make the social network co-operative and stable. By furnishing information on people within one's vicinity, ***SapienNet***, deployed on hand held devices and in the future, on the more ubiquitous mobile phones, will constantly strive to bring people, who need each other, closer, both socially and practically. As pointed out earlier this system will be of vital importance to people who are in dire need in alien lands. Pertinent social as well business communities, can be formed effortlessly. While a normal human could tweak the system to cater to a wide variety of applications, the physically challenged can make use of the same to find whether help is available in his/her vicinity. On the whole this networking system will help people come closer and work to build a better society.

3. IMPLEMENTATION & ENGINEERING CONSIDERATIONS

3.1 User Scenario

To gauge the versatility and the effectiveness with which SapienNet can spontaneously stimulate a user to venture beyond social and communication boundaries, consider this real-life scenario -

James, an employee of a software concern, has been assigned the job of receiving an official guest Clive at the airport. James as such is unaware of the looks of the person he is supposed to host. On reaching the airport, James finds the flight to be delayed due to bad weather. Equipped with a PDA with SapienNet, James switches on an ActiveAlert and stretches out on a sofa near the Arrival lounge. While idly looking around his surroundings, he catches a glimpse of a face that looks familiar but is unable to place. Not wishing to embarrass himself by walking over and introducing himself, James decides to get information on the person. He clicks on the SapienMap and is elated to discover that the person is an old long forgotten buddy, Henry. One more button click and he gets complete information on his buddy, where he works, his address, profession, etc. Energized with the excitement he forges ahead to meet Henry and an old relationship is revitalized thanks to SapienNet. Both friends catch up with on each other's lives and bookmark one another using the SmartBookmarking application. While sipping hot coffee at the airport, James is taken aback by the sight of an elderly man who has all of sudden collapsed. He remembers his digital assistant again and plunges into action. Using PeopleSearch he quickly submits the query to search for a doctor in his vicinity. The presence of a Doctor is confirmed by PeopleSearch and his phone number delivered. James calls up the doctor on his mobile phone. By now PeopleSearch has already delivered a digital image of the Doctor enabling James to beckon him towards the patient. James then proceeds to check the medical history of the elderly man using SapienMap and PeopleSearch only to find that he is a heart patient. James informs the Doctor about this and first aid is delivered and an ambulance called. James completes his duty by making a final call to the family informing them of the incident. Thanks again to SapienNet for having provided the family (emergency) phone number of the patient and of course for the timely aid in saving a life. The announcement of the arrival of the flight is heard as also a beep from the ActiveAlert application indicating the arrival of the guest. James zeros in on the coordinates of the guest on the SapienMap. He uses PeopleSearch again and lo and behold he has all information of the person including his favorite cuisine. A sumptuous meal at a Thai restaurant and the three go back to routine work. James finishes his day as a dutiful employee, a good Samaritan and man who found a long lost friend courtesy of SapienNet.

This scenario demonstrates the possible circumstances and the role played by SapienNet in breaking the communication barriers, induced by factors like human diffidence and social stigmas.

SapienNet basically comprises of four basic components –

- A ***Social Component*** comprising of the people involved.
- A ***Hardware Component*** possessed by each individual in SapienNet.
- The ***SapienNet Server*** that hosts the information about all registered users and relevant web services.
- A ***Software Component*** in the form of application programs running on each of these devices.

3.2 The Social Component

A naming scheme henceforth referred to as the ***Human Identification Number*** (HIN) is given to all those who register at SapienNet. In the real world, a governing agency may decide the policy for allocating these numbers. A new user has to initially procure an HIN from this governing agency. The agency provides the user with a hardware authentication module (HAM). A HAM has the associated HIN hardwired into it. In order to authenticate a user, the HAM is used by the SapienNet system software running on the hand held device. The new user has to provide correct *a priori* information that is then stored in the SapienNet server's database. To ensure that the information supplied is correct, the governing agency could demand a copy of the passport, or driving license before issuing the HAM. With this HAM the user can join the community of SapienNet users.

3.3 The Hardware Component

Each SapienNet user possesses a GPS equipped PDA with integrated 2-way wireless capabilities (GSM/GPRS and WLAN - 802.11).

3.3.1 PDA and E-Box II

This contest entry appears for both IEEE CSIDC and Microsoft's Windows Embedded Student Challenge (WESC). WESC provided each registered team with the **E-Box II Thin client**. This E-Box had to be used to represent an embedded device that was being used in the project. We have used the E-Box to represent the PDA each user possesses. *The E-Box has a hardware configuration very similar to a PDA* (in terms of Processor clock, onboard RAM, Flash Memory, etc.) and runs on Windows CE.

Processor	Vortex86 200MHz Processor
Operating System	Microsoft® Windows CE
Memory	128MB onboard DRAM; 32MB IDE Type Flash Memory DOM- 32MB)
Ports Available	3 USB Ports, PS/2 Keyboard, PS/2 Mouse, 1 Parallel Port, 1 Serial Port, 10/100 BaseT LAN

Table 3.1 Specifications of E-Box II Thin Client

Table 3.1 summarizes the specification of the E-Box while Table 3.2 shows that of a typical Pocket PC.

Building and deploying a customized Windows CE OS image onto the E-Box was done using the Windows CE .NET Platform Builder 5.0 provided along with the E-Box SDK. A Windows CE image for the Vortex:x86 Board Support Package (BSP) was developed. (A BSP contains a set of device drivers that are added to the image design. In this only the required components are included.) Additional drivers were added to this image from the *Platform Builder Feature Catalog*, to enable support for special hardware components. This image was then built and installed on to the E-Box. Similar work was done to form the images which were to be run on the Laptops running Windows CE which we deployed.

Processor	Intel® PXA263 400 MHz processor
Operating System	Microsoft® Mobile Software for Pocket PC 2003 Premium Edition
Memory	Main Memory: 128MB SDRAM Program Memory: 32MB CMOS Flash ROM; Application Memory: 32MB NAND Memory (Flash ROM Disk)
Connectivity	Integrated Wi-Fi (IEEE 802.11b)

Table 3.2: Specifications of Toshiba PocketPC

Since a multi node ad-hoc network was to be created, **a TOSHIBA e800 PocketPC** and **two CEPC Laptops** were also used. CEPC Laptops had the custom Windows CE image running on them.

3.3.2 Wireless Adapter

Wireless connectivity is required for the formation of ad-hoc networks of SapienNet users in a SapienZone. The E-Box does not have a wireless interface. For wireless connectivity, we connected an **Ethernet**-to-**Wireless Bridge** [D-Link DWL-G810 Enhanced 2.4GHz (802.11g)] to its Ethernet adapter. To make the adapter work, the Windows CE Ethernet driver was added to our custom Windows CE image. The Toshiba PDA and the laptops had built in wireless connectivity. The wireless interface can also be used to connect to the Internet if there is a Wireless Access Point (WAP) within range.

3.2.3 GPS Receiver

A GPS Receiver is used to get the current location of a user. This location information is sent to all other users within the *SapienZone.* This makes the ad-hoc network location-aware. We have used the Garmin 18 GPS with USB interface. Garmin does not have a device driver for Windows CE. Therefore, Garmin's SDK for USB programming was used for interfacing with the Windows CE E-Box. Garmin GPS outputs data in its own proprietary format, which happens to be a drawback of this device. They do not follow the industry standard NMEA 0183 data format. However the proprietary format is an open one and can be parsed easily. Garmin's product was used since no other GPS receiver was available at IIT Guwahati. Again Windows CE USB driver support has to be added to the custom CE image. The SapienNet system software requires the GPS to be always on. This leads to continuous power consumption, and the PDA might have to be re-charged frequently. A solution could be a self-powered GPS receiver.

3.2.4 GPRS Modem

The GPRS modem is required for connecting to the Internet using a cellular network backbone. GPRS is a standard solution for mobile internet connectivity. Internet connectivity is required since the SapienNet system software running on the PDA consumes web services. The Wavecom GSM/GPRS Modem with RS232 interface is being used. We have interfaced the GPRS modem with the E-Box. However, there is no GPRS service provider in the IIT Guwahati area. Therefore its testing will be completed in another region which has GPRS connectivity, in the last stage of the project development lifecycle. (Currently the wireless interface is being used to connect to the web services through a WAP). For the GPRS modem, Windows CE serial port driver has been added to the CE image.

3.2.5 Hardware Authentication Module (HAM): External non-erasable memory

In the proposed setup each embedded device is referred by a unique identifier called the ***H**uman **I**de**N**tifier* (HIN). It is used for identifying each person uniquely on the SapienNet. It is mandatory that proper security procedures be used to validate the authenticity of the HINs. This is done by storing an HIN in an external non-erasable memory. We have used an EPROM interfaced to the E-Box via the parallel port for this purpose. It is proposed that the HAM should be provided by the governing agency to each new user. Support for the Parallel port was also included in the Windows CE image.

3.3 SapienNet Server

The SapienNet server comprises of an IIS http server and an Oracle 9i database server. The IIS server hosts the web services. A database containing information about the users of SapienNet is hosted on the Oracle9i database server. Whenever a user registers with SapienNet, his/her information is stored on to the database. The design of this database is discussed later. The web services handle requests from the client PDAs. The web service in turn sends queries to the database to retrieve the required information. The web methods that a client PDA can invoke are explained in the section 3.4. A Visual C# ASP.NET web service was deployed. The use of .NET architecture is of great advantage here since consuming an ASP.NET web service through a Windows Application is fairly straightforward. By adding a web reference to the URL where the web service is hosted the consumer can access the available web methods. In our case the consumers were CE.NET windows applications. Web services technology was chosen as it offered interoperability since it uses open standards (based on XML) to transfer data. They are self-contained, self-describing, modular applications that can be published, located and invoked across the Web and hence were found suitable.

3.3.1 Database Design

The database comprises of a set of tables which contains all the information about a person. This set contains the table comprising of the essential fields which all users are required to fill in while registering with the system .The other tables contain the remaining fields which are optional. This division of the user's information into the *mandatory*, *likely* and *least-likely* tables helps in minimizing storage wastage, as some users may not fill in any values for *likey* and/or *least-likely* table values.

Apart from this information, we use another table to grant two levels of access over each field of the user's information. This table contains the HINs of all the other users marked as *friends*. The fields of the first set of tables may be either open to be accessed by everyone or they may be constrained to be shown to only *friends* of requesting HINs. We represent the access level of each field of the user's information by a binary digit. A zero may be used to represent a free-to-all access level while a one may correspond to access to only *friends*. Hence the access associated with all the fields can be represented by a string of bits with length equal to the total fields in a person's information.

3.4 The Software Component

Each PDA runs the SapienNet system software. This software consists of a network component termed as the *HINGrabber* and an application suite. This application suite consists of five applications: *PeopleSearch, SmartBookmarking, ActiveAlert, EarlyIntro* and *SapienMap*.

The *HIN Grabber* maps all the HINs and the co-ordinates of other users in the vicinity of the device (SapienZone) to the application layer. The *PeopleSearch* program uses these socially meaningless HINs and converts them into useful information by connecting to the web services on the SapienNet server and retrieving data from its database. When a user is not too sure of whether to commence a verbal conversation with a person in his/her vicinity, *EarlyIntro*, another application program, aids him in taking the right decision by providing pertinent information. *ActiveAlert* extends the concept of *PeopleSearch* by alerting the user as and when the person of his choice comes within his SapienZone. Thus if *PeopleSearch* initially reported the absence of a Doctor in the user's SapienZone, it raises a signal the moment the Doctor enters it. *SapienMap* uses the co-ordinates obtained from the *HINGrabber* to draw a live map of users all around. *SmartBookmarking* could be used to bookmark people of interest.

3.4.1 HINGrabber

HINGrabber is an application, which is responsible for maintaining the HINs of all the nodes in the vicinity along with the position co-ordinates of the respective nodes. The data structure maintained by this application for HINs to position mapping is used by the other five applications.

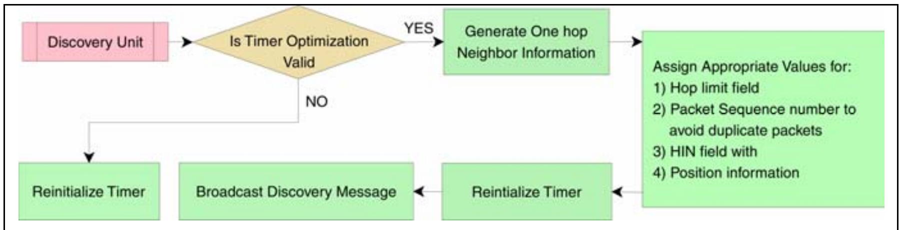


Fig 2: *The Neighbor Maintenance Algorithm.*

HINGrabber is a combined server-client application which periodically broadcasts packets using socket datagrams and hears to other broadcasts. It broadcasts two types of packets. One is called the *hello* packet and the other is called the *announcement* packet. *Hello* packets are used by nodes to inform its immediate (one-hop) neighbors of their existence, while the *announcement* packets are used to inform up to k-hop neighbors, the same. Therefore, only the users in this hop limit can listen to its announcement packets and vice versa and thus defines the vicinity for that user which we have termed as SapienZone. The structure of *hello* packet and *announcement* packet are shown in fig. 4.

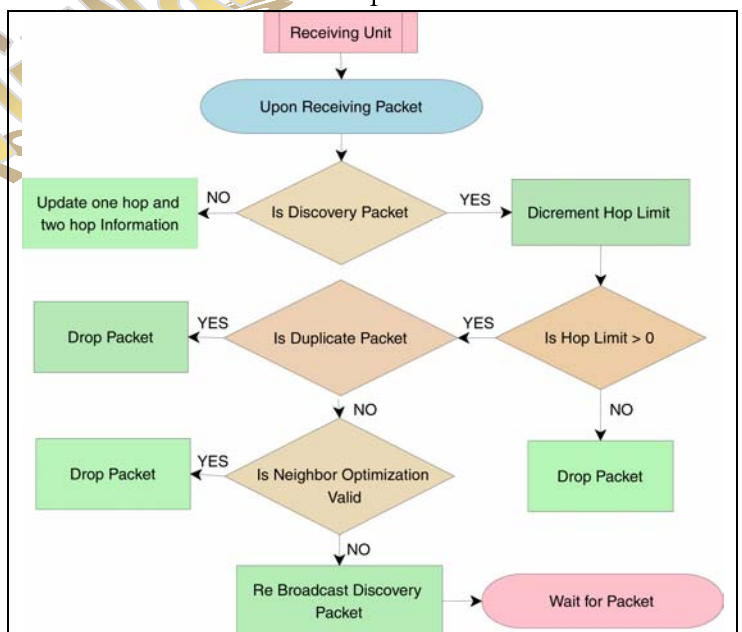


Fig 3: *The receiving side handling for the broadcasting unit.*

The biggest problem by deployment of such a broadcast application in a network is the excess of redundant broadcast packets. To reduce the broadcasting significantly, we developed and employed the two algorithms ***Timer Optimization*** and ***Neighborhood Optimization***, which are discussed later in the section.

Whenever a node receives a packet, it first classifies the packets based on whether they are periodic *hello* packets or *announcement* packets. If they are *hello* packets, they are used to update the neighbor table, with the HIN of that node, hops as one and the new timestamp. If a node finds a packet as an *announcement* packet, it reduces the *hop limit* field by one and checks if it is a duplicate packet through the assigned sequence number. If the packet is not a duplicate packet, it checks if the *hop limit* > **0**. If a packet passes through both the checks, it finally checks whether the rebroadcast is really required or the rebroadcast is redundant through the ***Neighbor Optimization*** algorithm. If the ***Neighbor Optimization*** algorithm allows a packet, then it is re-broadcasted in the network. This algorithm is explained as flowchart in the Fig. 3.

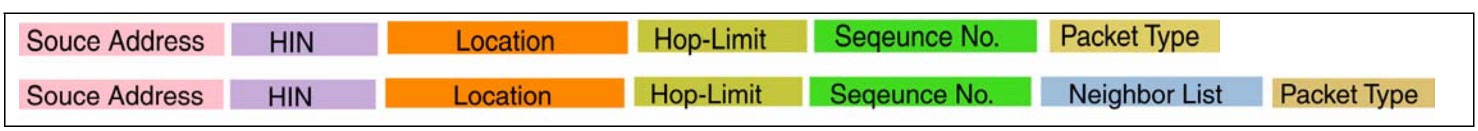


Fig. 4 Structure of *announcement* and *hello* packet

Timer Optimization Algorithm

To determine the time for sending multi-hop discovery packet to the nodes in proximity, we have utilized the ***distance effect***. ***Distance effect*** [8] means that greater the distance separating two nodes, the slower they appear to be moving with respect to each other. We have used this idea to optimize the frequency at which a node sends announcement packets in the network and thus correspondingly saves on the bandwidth and energy consumed. Thus, we keep one hop information continuously updated using the *hello* packets, and use this one-hop neighborhood information to make the decision to send the announcement packet.

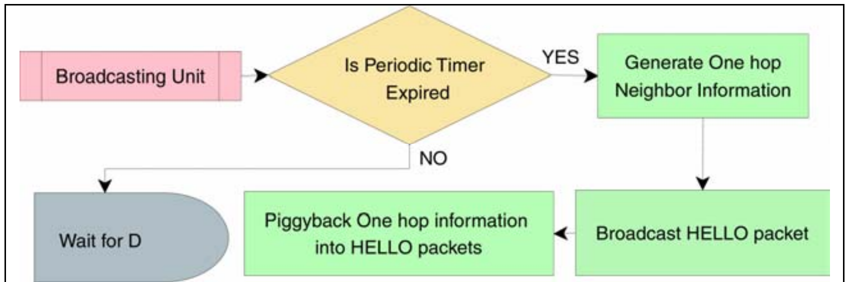


Fig 5: *The Timer Optimization Algorithm.*

To explain the timer optimization algorithm, we denote the set of nodes in the neighborhood of a node ***u*** at time ***t*** by ***N(u, t)***. We have introduced the time factor to take into account the network dynamics. Thus, in this optimization, if $| N(u, t) / N(u, t-1) | < N_{threshold}$, then the *announcement* packets are not sent. In other words, the Timer Optimization Algorithm can be described as follows: *All the nodes periodically gather information about neighbor's one-hop away. If the node finds that the number of neighboring nodes that has changed is higher than a certain threshold, it performs network-wide broadcast based upon an assigned hop limit field to the packet.* This optimization adaptively controls the timer; saving resources of the participating nodes.

Neighborhood Optimization

Neighborhood optimization is based on the optimization used in OLSR [5]. It utilizes the fact that a second-hop neighbor may be reachable from more than a one-hop neighbor. Thus each node selects only a subset of one-hop neighbors who forward the *announcement* packets further. We call this selected subset as the *broadcast set*. We use the following heuristic for the calculation of *broadcast set*: *Initially, we assume that broadcast set is empty. Then we add all the nodes in one-hop neighborhood that are the only path to some second-hop neighbors. Then we add all the nodes in one-hop neighborhood, which provide the link to the maximum number of non-reachable second hop neighbors. We keep on adding nodes, until all second hop neighbors are reachable. In case of multiple choices, we select the node that is reachable to most number of nodes in the second hop. The nodes selected as subset nodes are informed through hello packets that they have been chosen as broadcast nodes.*

3.4.2 PeopleSearch

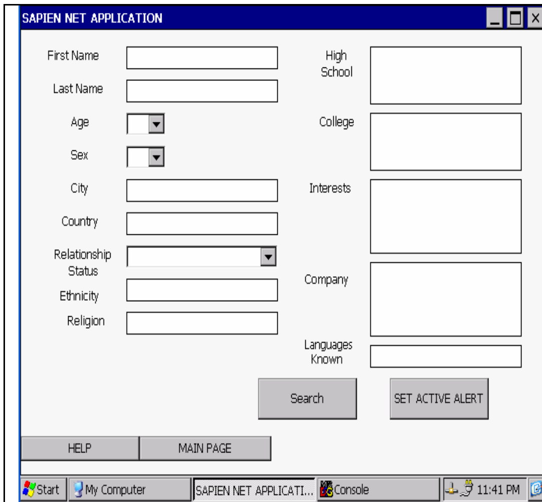


Fig. 6: The User Interface for PeopleSearch Application.

PeopleSearch helps in searching, identifying and locating desired persons in a non-intrusive manner in the user's SapienZone. The user enters values into the fields of the *PeopleSearch* query form. The values specified are then matched with the a priori information of the people in the vicinity. The results from *PeopleSearch* supply details about the people who fit the desired profile. The user can subsequently find out more about the person of interest; get more details and a photograph from the SapienNet server.

The application *PeopleSearch* interacts with the Web Service hosted on the IIS Server. A web method (called *searchPeople*) is invoked by this application, and 3 arguments are passed to this web method. The first argument contains the desired characteristics demanded by the user; the second argument contains all the HINs available in the vicinity (obtained from *HINGrabber*). The third argument is the user's own HIN. Using these arguments a query is sent to the Oracle 9i database server by the web service. The web method returns a list and specific details of the desired people. These details are then displayed by the *PeopleSearch* application.

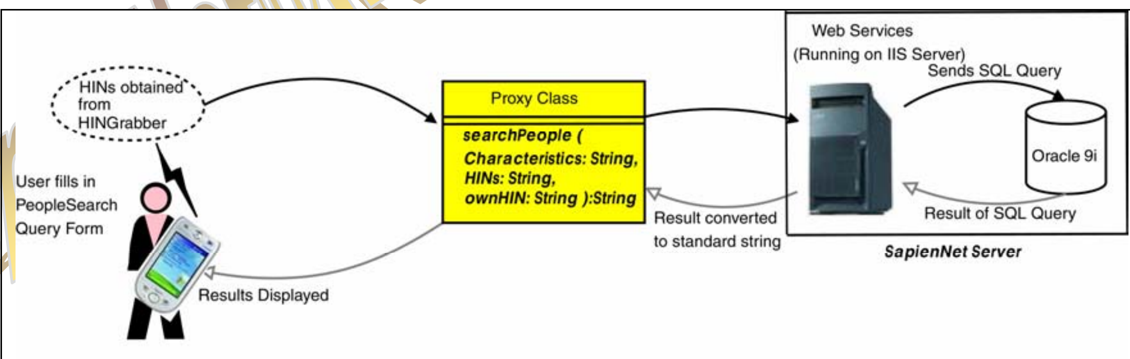


Fig. 7: Shows the basic mechanism of the *PeopleSearch* Application.

The web method *searchPeople* returns only the most important details and not all the details about the desired people to avoid sending too much data over the network. Since *searchPeople* can return details of a lot of people, getting all that details of these people won't be quite useful. When more detail of a specific person is required by the user, he can use the facility available, for getting more information. The drawback of such a scheme is the invoking of the web method *getDetails* every time more details of a person are required.

Web Method	**Description**
`public string searchPeople(string characteristics, string HINs, string ownHIN);`	Used for searching desired people in the vicinity.
`public string getDetails(string HIN, string ownHIN);`	Used for getting complete details of a person whose HIN is known.
`public string getPhoto(string HIN, string ownHIN);`	Used for getting the digital image of a person whose HIN is known.

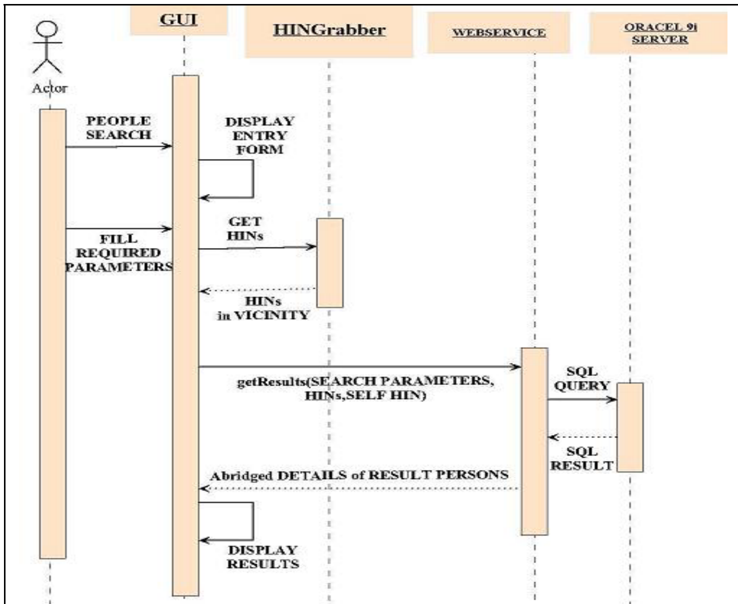


Fig. 8: Sequence Diagram

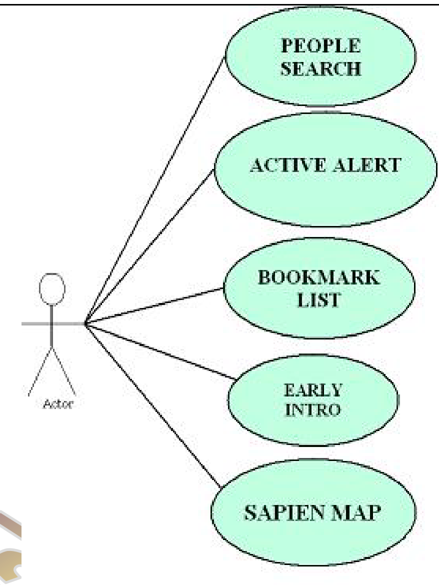


Fig. 9: Use Case with actor

3.4.3 Smart Bookmarking

SmartBookmarking is an application which allows the user to bookmark individuals he feels are of interest. For example, one could bookmark people who appear as results from *PeopleSearch*. These bookmarks can be used for later reference. The bookmarks are stored locally on the PDA as well as the SapienNet Database using the *bookmark* web method. Few important details of the person bookmarked are kept locally. Again the web method *getDetails* and *getPhoto* are used to obtain complete information of the bookmarked person. On the server, for each user a list of bookmarked HINs is kept. The PDA can get the bookmark list from the SapienNet server using the web method *getBookmarks* if required (in case of data loss on the PDA).

3.4.4 Active Alert

Web Method	Description
`Public string` ***bookmark***`(string HIN, string ownHIN);`	For storing the bookmark on the SapienNet database.
`Public string` ***getbookmarks***`(string HIN, string ownHIN);`	For obtaining the list of bookmarks from the SapienNet server onto the local PDA.

ActiveAlert is a trigger that can be set by the user for real time alerts to detect whenever a desired person comes within vicinity. The user sets a timer, during which *ActiveAlert* keeps looking for the person in the vicinity. If the desired person is located in the vicinity within the set time then an alert window pops up, otherwise the timer expires. The user can set the *ActiveAlert* in 2 ways. One is by setting *ActiveAlert* on a bookmarked individual(s) or by defining characteristics (as done in *PeopleSearch*).

ActiveAlert for the bookmarked individual compares the HIN of the bookmark (available locally on the PDA) with the HINs available from *HINGrabber*. This comparison is done every time the table of HINs changes due to *HINGrabber*.

ActiveAlert in which search characteristics are specified, the web method *searchPeople* is re-invoked at a pre-determined refresh rate. This is done till a suitable match is found or if the timer expires.

3.4.5 Early Intro

Web Method	Description
`Public string introduce(string HINs, string ownHIN);`	For obtaining information about every user in the vicinity.

EarlyIntro helps diffident users to explore and search for more information on specific people in his/her vicinity before entering into a verbal conversation with them. This will help in shedding social inhibitions to quite an extent. Here the user requests for details of every individual in the vicinity. He is provided some important details about every person in the vicinity. He can subsequently choose to get more information about certain individual(s).

The web method *introduce* is invoked with a two arguments, first is the list containing all the HINs in the vicinity, second being the user's own HIN. The web method returns data just as the web method *searchPeople* does, and when the user chooses to find out more, the web method *getDetails* is invoked.

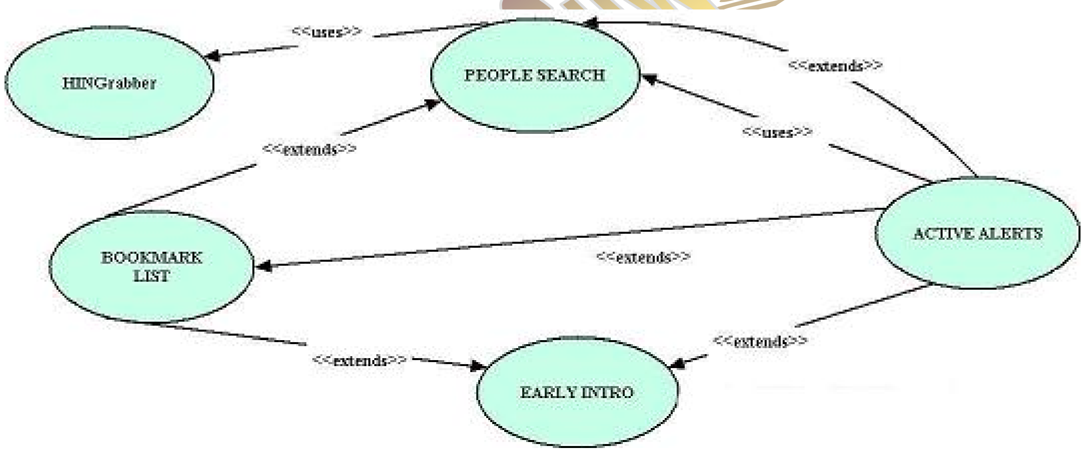


Fig. 10. Use Case Diagram

3.4.6 SapienMap

The *SapienMap* application gives a live map of the users in the vicinity. The design of the *SapienMap* interface is an intuitive one. It includes support for the user to look up more details about a person showing up on the map by clicking on it. This functionality is provided by using the *getDetails* web method. *SapienMap* was implemented using the GDI+ API. For rendering the graphics on the .NET Compact Framework two APIs were considered: GAPI or GDI+ API. GDI+ was chosen because GDI+ is a native Win32 API while GAPI does not have pre-built support in the .NET Framework.

3.5 Some Engineering Considerations

3.5.1 GPRS versus WiFi

Although access points are increasing in number, but they are still not ubiquitous, and network always internet connectivity using WiFi is still a distant dream. Hence, we used GPRS cellular infrastructure, to provide internet connectivity to the users, as this is the best available mechanism currently for un-interrupted internet connection for mobile users.

3.5.2 Choice of Development Platform

For development of these applications for Windows CE .NET we considered two options: Win32 and .NET Compact Framework. The .NET Compact Framework API (Visual C#) was chosen over Win32 (Embedded Visual C++) in designing the GUI of *PeopleSearch, ActiveAlert, etc* unlike the application *HINGrabber*.

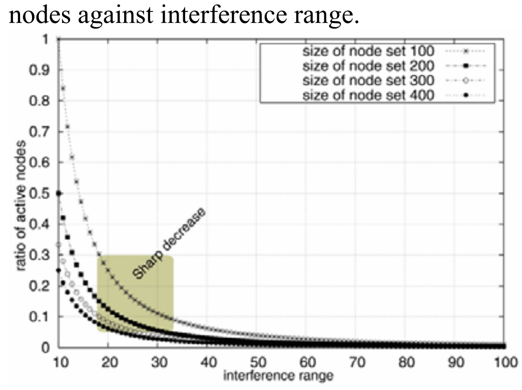
Advantages of .NET Compact Framework	**Advantages of Win32 API**
The foremost advantage of the .NET Compact Framework is Rapid Application Development. Drag and Drop controls used for quick development of the forms used for the GUI.	Win32 API applications run extremely fast because Win32 executables ship as native machine instructions.
The .NET Compact Framework is excellent for developing user interfaces, web service clients, for data intensive applications. (Win32 has no built-in container classes.)	Win32 API used for creating low level code for device drivers, OS support, real time support etc. (Hence chosen for *HINGrabber).*

The conclusion was that since the applications *PeopleSearch,* etc are high level applications and act as web service clients it would be better to use the .NET Compact Framework. Also, these applications do not have stringent real time response requirements and are slightly data intensive. Additionally, .NET Compact Framework application can call Win32 dynamic link libraries, including the system's core library, COREDLL.DLL, by using a feature called "Platform Invoke." Hence while developing applications which use the .NET compact framework, some Win32 API functions can be used when required.

3.5.3 Multi-hop Network versus Single hop Network

The first question that can be raised is why to use multi-hop ad hoc communication. The first reason that multi-hop ad hoc network is that it ***do not require existing infrastructures***. Further, ***single-hop communication significantly decreases the capacity***. Assuming if a node is active, no other node can be active within a range R_{int}, called as the *interference range*, of area $A_{int} = \pi R_{int}^2$. Let N_{int} is the maximum number of nodes that can be simultaneously active and N be the total number of nodes. If we consider network to be sufficiently dense, the fraction of active nodes, $\rho = (N_{int} / N) = (1 / N) (D / 2 R_{int})^2$. Fig. 11 shows the affect of using long range broadcast. The setup uses a network of radius **100** *m* with interference range between **10** *m* to **100** *m*. It is clearly observed that after **20** *m* of interference range, ρ reduces to ***30%***. We can conclude that through multi-hop communication, we can increase the capacity and hence better performance.

Figure 11. Shows the variation of fraction of active nodes against interference range.



3.6 Tools Developed

For the purpose of testing and verification, we developed a Network Simulator and a GPS simulator.

3.6.1 GPS Simulator

During the initial phase of the design, the GPS receiver set was not available. To continue the development without GPS, we implemented a simulation tool for GPS. Random Walk Mobility Model [14] was used as the mobility model to generate the random location values, which was output in the Garmin proprietary format.

3.6.2 Network Simulator

An event based simulator for scalability analysis was developed in C++. In the simulator, each mobile node has two states, *move* or *stay*. When it decides to move it chooses a randomly selected point and moves towards it at a constant speed. After the mobile node has reached the position, it stays in there for a randomly chosen amount of time within the maximum sleep time.

For evaluating connectivity, a simulation area of *200 m × 300 m* is considered. The transmission range is varying in first case and in next case fixed at *80 m*, node speed of *5 m /sec* in first case and varies in second case, max sleep time of *30 second*, total number of stations 25, and simulation period time *500 second*. The first parameter is the network connectivity, the percentage of the total simulation time in which the network remains connected. Fig. 12 shows connectivity against transmission range. It shows the advantage of using higher transmission range, especially for the case of *60 m* to *70 m* range case. This shows the need of using an optimal transmission range since lowering down transmission range for energy preservation can badly affect the network, resulting in partitions. Fig. 13 shows connectivity against network mobility. Mobility is varied from 2 m/sec to 12 m/sec. It shows that connectivity is not affected by node mobility.

Fig. 12. The variation of transmission range against the network connectivity.

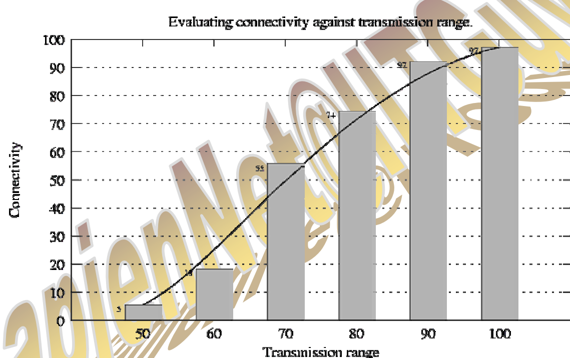
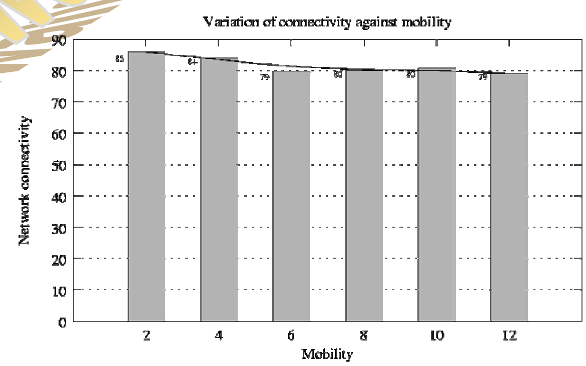


Fig. 13. The variation of network connectivity against the transmission range.



3.7 Current Work Status

Currently we are in the final phase of development. Until now, we have completely implemented and tested our messaging algorithm and the complete application suite which includes *PeopleSearch*, *ActiveAlert*, *SmartBookmarking EarlyIntro* and *SapienMap*. The customized optimal image of Windows CE required for our project has also been configured, deployed and tested on the E-Box for its correct functioning and support for our applications. Web services and the Oracle 9i database server has also been deployed and tested alongside our applications. The tools developed for testing (GPS Simulator and Network Simulator), have also been completed. All these developed modules have been tested on a 4 node network which includes an E-Box, a PDA and two Laptop CE PCs which access the web services via a wireless access point which emulates the functionality of GPRS connectivity.

We have identified the exact procedures which we have to do to complete the project, and expect this phase to last for 3 more weeks. We have already started the GPS interfacing by using the USB SDK provided by Garmin, for directly accessing their USB enabled GPS. GPRS has already been interfaced, but due to the lack of GPRS connection availability in our location, we haven't been able to test it. After the authentication module has been implemented and tested, we will move to a neighboring location and test the GPRS connection along with the entire work.

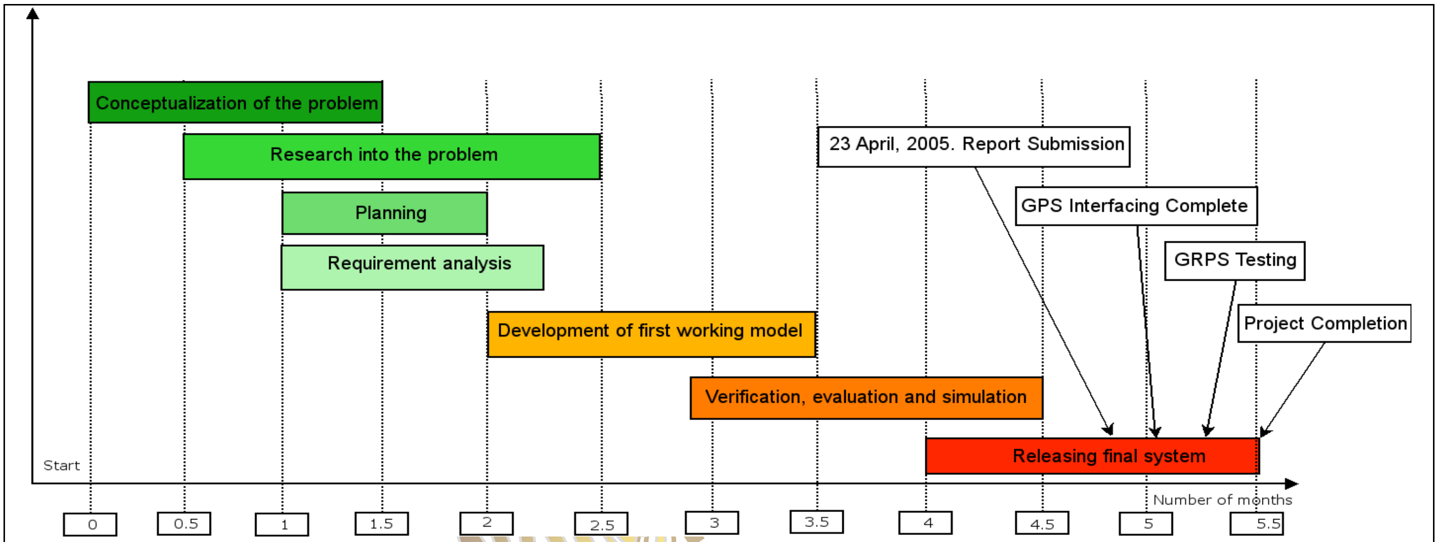


Fig. 14. The Timeline Diagram

3.7 Verification and Testing

Verification of the system was done using the tools developed, the generated scenarios and inputs based on an incremental approach. In the first step, we mathematically verified our decision of using a multiple hop network, which has been discussed in the tradeoff section. Then, using the simulator as discussed above we concluded that the connectivity is not affected by node mobility. These results were useful in acknowledging that the system can theoretically work in the envisaged environment. Then the web services was deployed and tested. It was followed by the deployment and testing of a CE PC. We tested that Windows CE functioned correctly, compiled and ran a few test programs on it. Then a whole setup of two CE PCs and a PC running Web Services and Oracle database server, networked through wire, was done to test the first beta version of our applications. Following the successful testing and the required debugging of the system, we installed the wireless cards on each CE PC and tested the system. Then, we tested our messaging algorithm after adding suitable test procedures within it, on a four node network communicating over wireless media. The four nodes used included an E-Box, a PDA and two CE PCs. Finally, the whole system was tested on this four node network along with Web Services running on a PC accessible via a wireless access point, which emulated the functioning of GPRS connection.

3.8 Feasibility and Marketability

SapienNet relies upon the support from mobile handhelds equipped with GPS, GPRS, wireless interfaces and an external hardware module. Recent market clearly indicates that mobile devices with such supports are available.

Mobile phones offering an integrated PDA and internet access with GPS support are available. One of the coherent examples is the feature rich iQue 3600, whose design is specially oriented towards business travelers and tourists. Regarding the location support, GPS provides two levels of service, a Standard Positioning Service (SPS) for general public use and an encoded Precise Positioning Service (PPS). SPS Coverage is continuous and worldwide, with a position dilution of precision (PDOP) of 6 or less which shows the feasibility of the system in terms of location support. Other examples include TF30-CF which can be used as an ordinary real-time GPS locator. A user can simply plug it into a PDA or other type of handheld computer running Windows CE operating system, with suitable mapping and routing software for navigation purpose. Many PDAs in the market come with an in-built smart card reader which can be used to read a smart card which can work as the external hardware module.

SapienNet has wide scope of finding success in market because the main emphasis in the design of SapienNet has been end-users. It has been designed to become an integral component of user's social life, by providing a variety of services as discussed above. SapienNet can thus be realized as a service provided by mobile phone service providers, and can serve as a key value-added service that telecom operators can offer.

4. Summary

4.1 Future Work

Our initial focus was on the basic requirement of the system. After accomplishing the above targets, we became interested in enhancing the capabilities of the system. The first improvisation, which we intend to consider, is the formulation of an efficient query through location-based user information storage. Since a user can get information about its current location status, its information can be dynamically maintained in location-based databases. Queries may be preprocessed and localized to target specific location based databases thereby reducing the cost of search. Among the other factors, we would also like to build a complete independent single-module device integrated with a GPS, GPRS and a Security system.

Another work to be dealt with is positioning support. On deployment, the indoor performance of the system can be affected since GPS performance degrades in closed operating environments. We plan to develop an adaptive locationing mechanism to which can be switch from GPS mode to another that uses indoor infrastructure for positioning, as and when we move into closed environments.

4.2 Conclusions

SapienNet augments the capability of mobile handhelds to deliver location-aware services to subscribers with spontaneous ad hoc networking and the positioning using a GPS. Such services can greatly improve our day-to-day life through better communication using the application suite developed.

SapienNet can be a key value-added service, which telecom operators can offer. To support efficient and effective development and deployment of innovative location-aware services to the user (like ***PeopleSearch, SmartBookmarking***), a flexible and resilient architecture has been built. This system was created to reap the benefits of better an enhanced communication among people, without compromising on their privacy. On the whole, the system has great potential and can provide significant improvements in communication among people, **dissolving the communication boundaries**.

Marketing the SapienNet service will be highly feasible since currently numerous mobile handhelds are equipped with GPS and WIFI. The boundary between mobile phones and PDAs is fast becoming blurred as more and more capabilities are being packaged into the former. The market has a plethora of PDAs equipped with multi-way connectivity options like Bluetooth, Wi-Fi , GPS and GPRS . One such instance is the HP iPAQ h6340. Such features enable the formation of a wireless ad-hoc network with other similar devices.

SapienNet was envisaged as a product that attempts to bridge the gaps that hamper seamless communication amongst people. These gaps are brought about due to the inherent nature of human beings and their reluctance to interact as also some social inhibitions.

5. References

[1] “Microsoft deploys its own search engine”, Web Site, Published on Mon, Jan. 31, 2005, http://www.broward.com/mld/siliconvalley/news/editorial/10783691.htm

[2] “The business-– and technology – of finding a friend in cyberspace”, Web Site, Published on January 26, 2005, http://www.newsforge.com/article.pl?sid=05/01/26/0111240

[3] Liang Cheng, “Service advertisement and discovery in mobile ad hoc networks”, *In Workshop on Ad hoc Communications and Collaboration in Ubiquitous Computing Environments, in conjunction with the ACM 2002 Conference on Computer Supported Cooperative Work*, New Orleans, Louisiana, USA, November 16-20, 2002.

[4] “An electronic ID card? Why and for whom?”, Web Site, http://eid.belgium.be/en/navigation/36072/index.html

[5] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laoulti, A. Qayyum and L. Viennot, “Optimized Link State Routing Protocol for Ad Hoc Networks,” *In Proc. of IEEE INMIC 2001*, pp 62 – 68.

[6] P. Abrahamsson, O. Salo, J. Ronkainen and J. Warsta, “Agile software development methods: Review and Analysis”, *In Espoo 2002*, VTT Publications 478, 107 p.

[7] A. Schäfer and W. Rechert,, “Security and Web Services: Specifications, Technologies and Frameworks”. http://www.ito.tu.darmstadt.de/publs/pdf/WSSPaper1.pdf

[8] S. Basagni, I. Chlamtac, V.R. Syrotiuk, and B.A. Woodward, “A distance routing effect algorithm for mobility (DREAM),” in *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom)*, 1998, pp.76–84.

[9] Charles E. Perkins, Elizabeth M. Royer: Ad-hoc On-Demand Distance Vector Routing, 1999.

[10] Jerzy R. Nawrocki, Bartosz Walter, Adam Wojciechowski: Toward Maturity Model for eXtreme Programming, IEEE Computer Press, Los Alamitos, 2001.

[11] Roger S Pressman: Software Engineering: A Practitioner's Approach, McGraw-Hill Science/Engineering/Math; 5th edition (November 1, 2001). **ISBN:** 0072496681

[12] Object-Oriented Project Management with UML – Murray R. Cantor – Danvers,MA, 1998 by John Wiley.

[13] http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dncenet/html/choose_api.asp

[14] Charles E. Perkins, Elizabeth M. Royer: Ad-hoc On-Demand Distance Vector Routing, 1999.

[15] Jerzy R. Nawrocki, Bartosz Walter, Adam Wojciechowski: Toward Maturity Model for eXtreme Programming, IEEE Computer Press, Los Alamitos, 2001.