

SPATIAL ALARMS

Anand Murugappan and Sarang Karandikar

{anandm,sarangk}@cc.gatech.edu

Abstract

In the present day world people frequently use what can be called temporal alarms i.e. time based alarms like *'Give me a wake up call at 8:00 AM tomorrow'*. These temporal alarms can be set on a variety of devices like clocks, PCs, mobile devices etc. Among all these, mobile devices have this additional feature that they are carried along by the person i.e. can move in a 2D coordinate system. With the advent of location tracking systems like GPS, mobile devices can locate their position in the coordinate system. So with a GPS enabled mobile device we can extend the idea of temporal alarms to space – there by introducing the concept of spatial alarms. To set a spatial alarm one would have to designate an area on the 2D coordinate system as a trigger area. When the mobile device enters the marked area the alarm would go off letting the user know that the spatial condition has been satisfied. This has several applications. Some of the obvious uses include applications like (a) *'Remind me when I am near place x to pick up stuff from y'* (b) *'Remind me not to go anywhere close to place x'* (c) *'Let me know if my friend x is within a mile of my current location'* etc. In this project we intend to focus on static spatial alarms i.e. alarms on fixed areas on the map. The focus of this project is on implementing a simple prototype system to demonstrate the viability of the idea. This document is to serve as the project proposal.

Motivation

All of us frequently use what can be called temporal alarms. Often we set wake up alarms and have the device (Clock, PC, mobile phone, PDA etc) notify that the time is up. Wouldn't it be nice to have a device remind us when we are within a particular area in the map? Such a system could be very beneficial. Several applications can be built on top of such a spatial alarm. To help demonstrate the usefulness – here are a few examples:

1. Spatial reminders: Assume that a person looked up a website like Craigslist and realized that a person at Marietta (a place near Atlanta) is selling furniture (something of interest). However the person would want not be interested in driving all the way to Marietta to check out the furniture. Instead he/she would want to be reminded the next time he/she drives within 1 mile of Marietta. As of now such a problem cannot be solved using temporal alarms simply because the user would not be able to exactly identify when he/she would be at that location.

2. Rendezvous: Another application could be in combination with social networking tools like orkut. Imagine you are in an airport and waiting for a flight. Also imagine that one of your friends is also in the same airport waiting for another flight. Wouldn't it be useful to have a system that could identify this and notify you 2, so that you catch up while you wait? Spatial alarms could be used for such purposes. All that one has to do is set a spatial alarm on the location of his/her friends (with their permission) and the moment the interested parties are within the trigger range (say 1 mile) they will be notified.

3. Spatial warning systems: Imagine facilities with specific areas marked as unsafe. As of now people use sign boards and barriers to make sure others know not to venture into the area. With spatial alarms this can be made even more effective. All that one has to do is install a spatial alarm on all those areas where he/she is to stay away from. Venturing into any of the areas by mistake will trigger of the alarm and he/she will be warned to stay away.

4. Inventory tracking: As of today, RFID and related technologies are used to efficiently use just in time inventory. We can also add another dimension to this by tracking the inventory movements by setting spatial alarms. For e.g., when the product reaches intended destination, the inventory management process can use this information to request for new resources and/ or optimize the current production queue. Even though RFID tags are cheap the RFID antennae are very expensive making this approach a lot more lucrative compared to RFIDs though it is just an approximation.

There are innumerable numbers of applications possible with such a framework.

Objective: In this project we hope to build a prototype system to show the viability of the idea. As a prototype we intend to implement a system with static spatial alarms.

Related Work

Spatial alarms are nothing but a specific case of Continuous Queries like those previously studied in literature like in MobiEyes [3]. The ideas in MobiEyes [3] describe how to achieve this on a generic query basis. In this case the trigger condition is always the same – ‘the mobile device entering the area of interest’. Though this is a special case it has special significance because a lot of applications can benefit out of this. In fact if this is offered as part of the mobile device platform framework application developers can come up with several useful applications like the one discussed in the motivation section. This idea also throws open several interesting research problems. One of them is as discussed in the last section. Though several research/articles efforts have talked of similar ideas we are not familiar with any software systems which are actually available for this task.

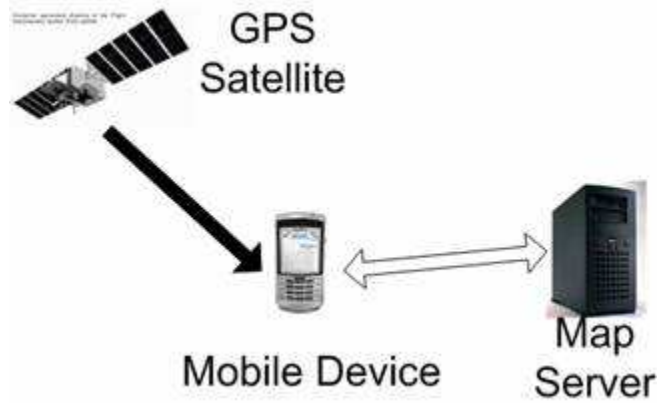
Proposed Work



User Interface

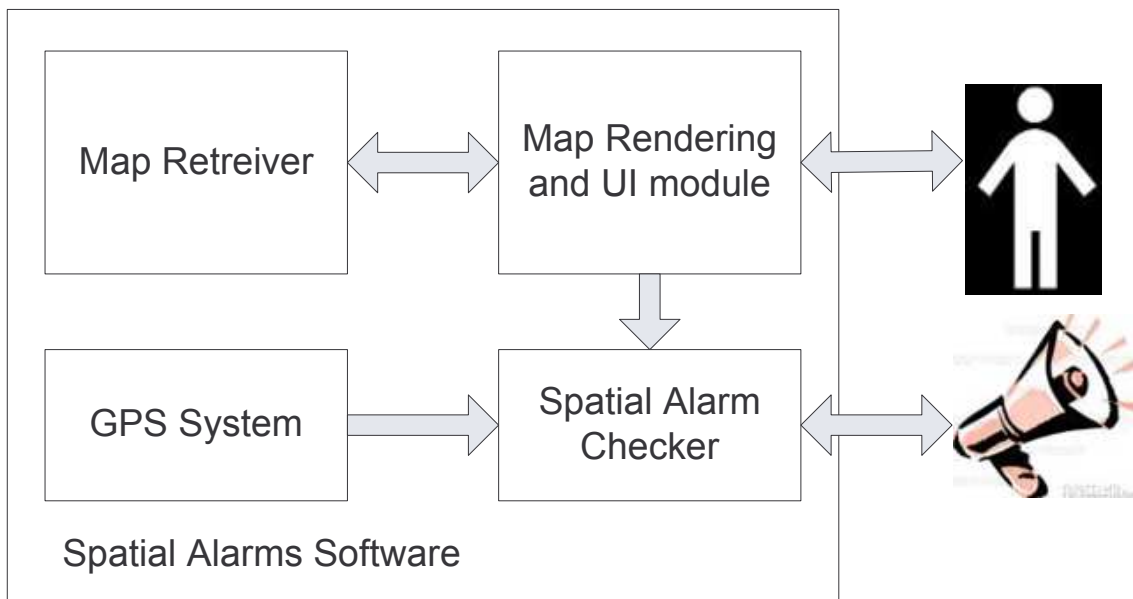
With regards to our estimation so far the following would be the resources required:

1. Mobile device/Emulator
2. GPS/Virtual GPS
3. Map service
4. Server to keep track of spatial alarms



System Overview

To start off the user would be able to see the map of the area and move around to focus on the desired one. The maps would be fetched from a map service like Microsoft MapPoint. The map would be displayed as shown in the figure above. The user then would mark the area where he/she would want to set the spatial alarm. Once the alarm is set the program would constantly check with the GPS if the spatial coordinates of the trigger area have been satisfied. Once satisfied the alarm would go off notifying the user. The User Interface seen in the figure below is taken from a Pocket PC 2002 simulator. Due to lack of current hardware, we would be using the simulator for demo. Operations on actual device should work the same.



The application would be installed on the mobile device. The application would fetch the map from the map server. The GPS device attached to the mobile device would get the coordinates of the device from the satellite to determine its position. The user would mark the trigger area on the map. The software would check periodically and raise the alarm if the trigger condition is satisfied by converting the GPS coordinates into a system that can be used with the user marked map coordinates.

Plan of Action

The following are the technologies we intend to use:

1. Visual Studio 2003 – C#.Net - We intend to write the bulk of the code in C# which has extensive support for mobile application development.
2. Web services – Map point [1] – We intend to use Microsoft MapPoint web service for fetching the maps.
3. Mobile Application Development toolkit – Microsoft has a mobile application development kit which ships with emulators and necessary libraries for developing mobile applications.
4. GeoFrameworks [GPS.NET](#) [2] – GPS dot net ships virtual GPS devices and necessary libraries for programming with the GPS. We intend to use the virtual devices here for this task.

Timeline:

1. Feb 16 – Completion of the proposal (this document)
2. Feb 23 (1 week) – Map rendering - learn how to use web services and fetch map from MapPoint
3. Mar 7 (2 weeks) – Map UI – learn how to make the map UI useful – User should be able to navigate and mark the area to set the trigger zone.
4. Mar 21 (2 weeks) – GPS – Fetch the mobile device location from GPS and integrate it with the system
5. Apr 4 (2 weeks) - Alarm activation – Based on the location indicated by the GPS system check if alarm has been activated. If so trigger the alarm (a beep or a message to the user).
6. Apr 11 (1 week) – Buffer & documentation – Should any of the above tasks take more than the estimated time this week will be used for wrap up! Also the final report will be made during this week.
7. Apr 16 – Completion of project (executable and documentation)

Deliverables & Evaluation/Testing:

1. Demonstration of static-spatial alarms. We do not intend to implement the dynamic spatial alarm part which would need far more work. User will be allowed to set only one alarm at a time. If time permits we will extend this to more than one.
2. Due to lack of hardware at this moment we intend to demonstrate **only on an emulator**. We will be using the Pocket PC 2002 emulator and [GPS.NET](#) for this purpose.
3. The evaluation should be based on how well we would have managed to achieve:
 1. Map rendering on the mobile device
 2. User interface to mark the trigger area
 3. GPS interaction module implementation
 4. The actual alarm on satisfying the trigger condition

Things that will not be addressed as part of this project:

1. Dynamic-spatial alarms i.e. spatial alarms on moving objects will not be considered.
2. All computations will be on the mobile device and not on any server and the program will run on the background. So should the pocket pc be reset the program will terminate and the alarm will no be valid. Of course this can be avoided by running a background service that could survive reboots but that would not be considered due to time restrictions. Anybody interested in something of this kind can always implement it using standard windows CE techniques.

Future work/ Potential research ideas:

1. For dynamic-spatial alarms there will be a need to have a logical central authority which would be responsible for identifying when the 2 moving objects are within trigger proximity. This would involve coming up smart server side efficient indexing solutions.
2. Implement an event based mechanism for alarm instead of poll based lookups.

References:

1. Microsoft Mappoint <http://msdn.microsoft.com/mappoint/>
2. [GPS.NET http://www.gpsdotnet.com/](http://www.gpsdotnet.com/)
3. MobiEyes: Distributed Processing of Continuously Moving Queries on Moving Objects in a Mobile System – Bugra Gedik and Ling Liu