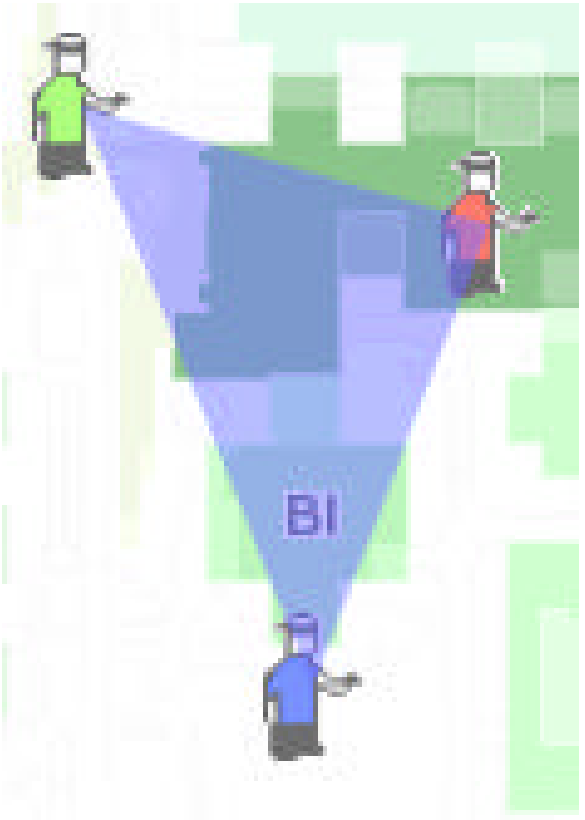
Building a *mobile*, *locative*, and *collaborative* application

by Fabien Girardin
Swiss Federal Institute of Technology in Lausanne
February 2005



Introduction

This document takes you through my journey in building CatchBob! [1], a treasure-hunt type, Wi-Fi based locative and collaborative mobile game. CatchBob! is used as part of Nicolas Nova's PhD research on spatial awareness at the Center for Research and Support for Training and its Technologies (CRAFT) [2]. Whenever I say "we" in this document it means Nicolas and I.

The idea to write this document came during a pervasive and locative workshop in London (PLAN [3], Pervasive and Locative Arts Network). I started to feel (like many in that workshop) that there was a momentum in the world of locative media. The pioneers in the locative mobile games have been experimenting their games (Uncle Roy All Around You [4], Can You See Me Now? [5], and others [6]) and talking about it for a couple of years. Papers have been written, techniques have been polished, tools have been developed, mistakes have been made, barriers found, and limits reached. It might be the time to demystify the complexity of creating mobile locative games. It is not a hacker's world anymore.

I first will introduce you the game itself, its purpose, the process we took to build it and its technical architecture. I, then, will talk about the data: what were needed and how to get them as well as what logs we were looking for and what we did with them. I will lead you into a more technical world as I explain how I implemented the positioning system and the client-server communication. I will then try to explain you what my experience was to build the user interface and what I learned from the users' perception. Finally, I will introduce a few topics relevant to the development of locative applications.

Hopefully this will be enough to eat and drink for you.

Overview

Location-based services recently happened to be one of the new cutting edge technologies in Computer Supported Cooperative Work. This project called CatchBob! focuses on its collaborative uses: how group of people benefits from knowing others' whereabouts when working together on a joint activity [7]. For that matter, we set up a collaborative mobile environment in which we test how a location awareness tool modifies the group interactions and communications, the way they perform a joint task as well as how they rely on this spatial information to coordinate. Series of experiments are run on this platform so that we can figure out how location awareness impacts group collaboration.

EXPERT SAY

Pasta & Vinegar *[36] is a resourceful blog run by Nicolas Nova around his research and interests in space, place, mobile computing and locative media.*

JARGON WATCH

Current best tools to develop locative media include ***Place Lab [37]*** *and* ***Mobile Bristol [38]****.*

JARGON WATCH

Awareness *is an understanding of the activities of the others, which provides a context for your own activity.*

Spatial awareness *is the appraisal and the understanding of information about the spatial positions of the partner(s) in the environment.*

Figure 1: CatchBob! players firing up their PocketPCs and planning a strategy to start the game.

[the game scenario]

We set up a mobile game in which groups of 3 team-mates have to solve a joint task. The aim of the game for the participants is to find a virtual object on our campus and enclosing it in a triangle formed by their position. The positions are provided by a location-based tool running on a PocketPC or a TabletPC. This tool allows each person to see the location of his or her partners as an avatar on the campus map. Another meaningful piece of information given by this tool is whether the user is close or far from the object: an individual proximity sensor. In addition, the tool also enables communication. On the PocketPC we provide a narrow channel of communication. If a participant points on a dot (that represents a person) with his/her stylus, (s)he can draw a vector that corresponds to a direction proposition for his/her partner: "go to this direction". On the TabletPC version the players have a broader channel of communication. They share their freehand annotation of the map.

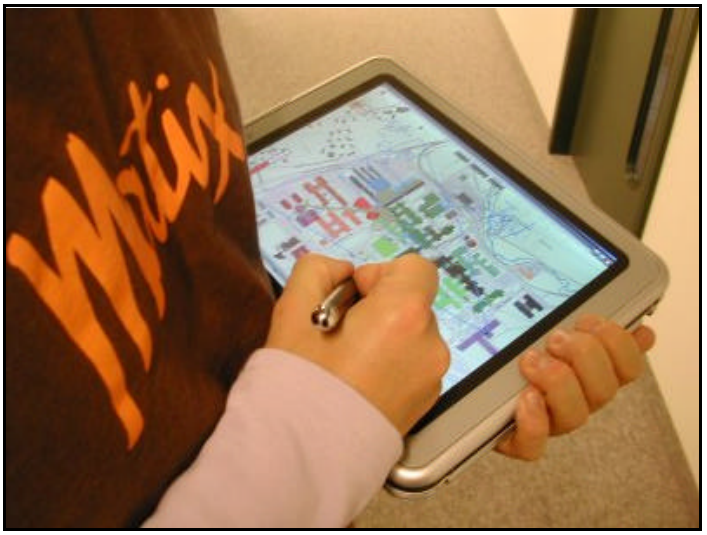


Figure 2: CatchBob! on TabletPC, a player annotating the map

[development process]

The development process of CatchBob! was rather classic. That is, it went against all proper project management methodologies. Joel on Software [8] is a must read if you are interested in properly lead a software project. We started with a very rough idea of game and technologies that could be used, poor requirements, and a fuzzy schedule. CatchBob! is roughly a 3 man-month, on and off, passionate effort. I built and tested it over a year using the 20% of my time allocated to locative media projects.

We based our approach on user centered and iterative design. Users are very creative; they often twist technology in unexpected ways, inventing practices before designers even think about them. We confronted colleagues and friends with semi-open implementations, observed how it was used and adapted it progressively through prototype-and-test cycles.

From a software engineer point of view, finding the right technologies and applying the proper techniques represented of course the core of the development process. Hallway usability testing ended up being very useful. However, it is more than average hard and time-consuming to setup proper test-cases for mobile and collaborative application. Especially when it comes to deal with unstable hardware and operating system functionalities. I also spent valuable time on making CatchBob! more tolerant to intermittent connections and latency. Proper quality assurance delivers trust. I must admit that it took longer than I thought until I could really feel confident about all the CatchBob! functionalities.

Part of the development process was to present our prototypes during workshops. After a workshop multiplayer games [9] [10] at the British HCI conference we came to the conclusion that our PocketPC version was providing a too narrow channel of communication between the players. It was so simple that not so many things could be conveyed with it. We moved to a TabletPC version with a bigger screen and easy freehand annotation.

[system architecture]

We use WiFi enabled PocketPCs and TabletPCs as clients. Each of them determines its location entirely privately without constant interaction with a centralized server. They use the Wireless Network sniffing capabilities of Place Lab or CRAFTDeamon to locate themselves by listening for Wi-Fi access points.

CatchBob! is built on a client-server communication model. Every 30 seconds, the clients broadcast their positions, commands and annotations via a centralized server. The communication is done over SOAP. Clients are in a pull mode in order to retrieve and synchronize the data.

EXPERT SAY

A lot of the very few people who have worked on really successful software projects go rich and retired to trout farms before they had a chance to pass on their accumulated experience to the next generation.

Joel Spolsky

JARGON WATCH

*In **user centered design**, the design takes in to account data gathered from the end users of the system.*

***Iterative design** is when the initial design is evolved and improved over many iterations.*

*A **hallway usability test** is where you grape the next person that passes by in the hallway and force them to try to use the code you just wrote.*

JARGON WATCH

***Place Lab** is software providing low-cost, easy-to-use device positioning for location-enhanced computing applications.*

***CRAFTDeamon** is my library for iPAQ that detects the MAC address and signal strength of nearby Wi-Fi access points.*

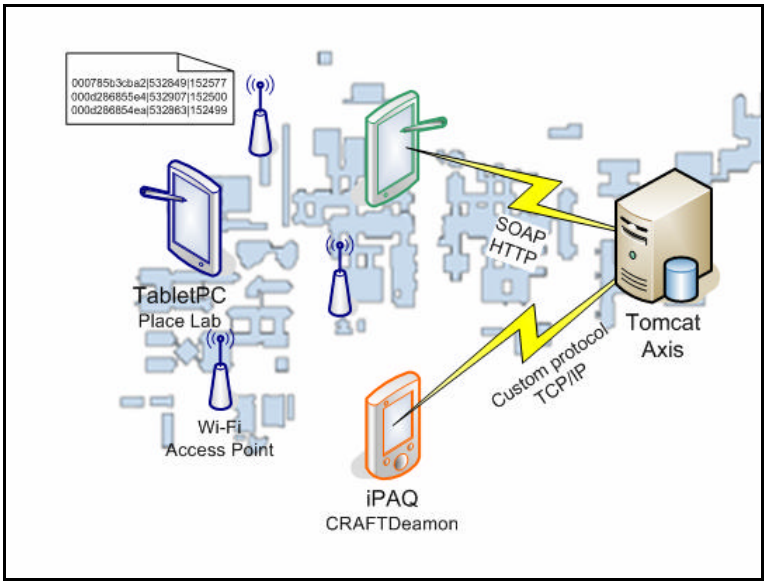


Figure 3: Positioning with triangulation on nearby Wi-Fi access points. Network sniffing is done with Place Lab on TabletPC and CRAFTDeamon on PocketPC. Data are transmitted to a centralized server via TCP/IP and my own protocol for PocketPC and SOAP for TabletPC

Some locative mobile games like the upcoming Yoshi Feeding, from Matthew Chamlers' team at the University of Glasgow, have now moved from client-server to plain peer-to-peer structure with epidemic algorithm. Even though I find this approach interesting for massive multi-user user turn-based mobile games, it applies less to a small 3-players game with a strong sense of synchronization and data consistency like CatchBob!. Moreover, for this game, it is just fine to have a single point of failure and potential bottleneck (the server). However, more massive locative and collaborative application can definitively benefit from decentralized and peer-to-peer systems based on RendezVous, JXTA, or Jini.

Data

[pre-requirements]

I had the luck to have at disposal all the data that I describe below. We mainly had to clearly and openly explain the usefulness of what we wanted to set up. Needless to say that explaining that building a locative game was part of our research in a swiss federal institute of technology raised some eyebrows (my euphemism). Nevertheless, I could get the hold of, the MAC address of the access points, their location (the name of the room), the spatial coordinates of the rooms and maps as GIFs (unfortunately no vector images).

JARGON WATCH

***SOAP** [39] is a simple XML based protocol to let applications exchange information over HTTP*

JARGON WATCH

*In an **epidemic algorithm**, the information is disseminated reliably in a distributed system the same way an epidemic would be propagated throughout a group of individuals: each process of the system chooses random peers to whom it relays the information it has received [40].*

JARGON WATCH

***Rendezvous** (named **Bonjour** now) enables automatic discovery of computers, devices, and services on IP networks.*

***JXTA** is a set of open protocols that allow any connected device on the network ranging from cell phones and wireless PDAs to PCs and servers to communicate and collaborate in a P2P manner.*

***Jini** is an open architecture that allows to create network-centric services that are highly adaptive to change.*

Figure 4: Wi-Fi coverage on in the campus in mid-2004. Map generated by knowing the position of the access points, not by net stumbling

If you don't have that luck, there are workarounds. I could have roughly drawn the map (e.g. based on an aerial picture of the area). With origin points outdoor, it is then easy to use a GPS and work with longitudes and latitudes. If the positions of the access points are not know, an empirical model can be used. That is, a radio frequency map of the area is created instead by doing some good old human self-positioning. The empirical model is explained in the "positioning techniques" section. This map stores the MAC address and signal strengths at each particular location from all access points that can be read from that location.

Tips and tricks are available in the upcoming O'Reilly's Mapping Hacks [11]. Wardriver communities [12] and freenetworks folks [13] have all the techniques and tools in hands. For example, Jo Walsh [14] and the people at Wireless London [15] are involved in making free-of-copyright street level maps of London walking, or cycling, the streets with GPS units, recording their traces and way pointing useful and easily identifiable points [16].

Not to forget that there are more simple Wi-Fi games around [17].

[security and privacy]

Networks administrators always expect the worst, and most of the time they are right when it comes to a 7000 people campus. However, making simple network auditioning does not represent any security breach. However, to us there is a clear privacy issue in tracking people. Even though the CatchBob! participants were willing guinea pigs using our hardware, we still had to make clear what we were building were not going to be used for other purposes. Clear communication is key, because there is a demand from people for security, privacy, and trustworthiness. The positioning technique used for my locative application such as CatchBob!

JARGON WATCH

Self-positioning *is when a user enters his/her own position to the system.*

EXPERT SAY

Taking GPS traces from bus journeys and bicycles, and using public domain sources of addresses to infer the vector semantics, we can easily build a free street map of some basic quality.

Wireless London

matches the privacy model of GPS. The devices compute location estimates autonomously rather than divulging their location to an infrastructure. Users know when they divulge their location.

[storing and logging]

I keep track of the players' present and past positions and annotations as well as their synchronization time. These data are stored in the centralized server's RAM. Mainly because the game lasts 1 hour and I trusted my server code to stay up and running (and indeed it never failed me!). I considered implementing some failover technique using Hibernate [18]. Hibernate is an Object Relational Mapper (ORM). It stores Java objects into a relational database. It is a great framework that keeps me away from JDBC. However I considered it overkill for my purpose.

JARGON WATCH

Hibernate *is a powerful, ultra-high performance object/relational persistence and query service for Java.*

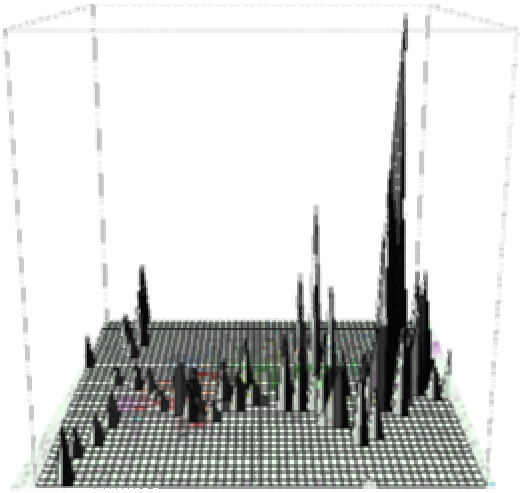


Figure 5: Based on the logs, analysis of the Wi-Fi usage during a CatchBob! game. The spikes is the spatial representation the number of refreshes players performed on their client's interface.

Nicolas wanted to find out how, when, and what kind of spatial information affect the individual and group performance in a joint task. Therefore, using log4j, I log the users' interactions with the interface; when and where they needed the other players' positions and when and where they annotate the map. The quality and usage of the environment the players move in is also grasped. Every 7 seconds, I log the nearby Wi-Fi beacons and the failed synchronizations to discover the seams and shadows of the playground. A synchronize event is fired every 30 seconds. It is more difficult to log the stylus strokes, because they are only arrays of x and y. For the analysis of the annotations, screenshots of each player's client is taken every 30 seconds. Here are examples of the formalism we used to log the games.

JARGON WATCH

Log4j *allows to enable logging at runtime without modifying the application binary*

Sample client logs

```
11:47:16,158|APs:2
```

```
11:47:16,158|000785b36281--81|000785b3653c--81|
11:41:22,839|STROKE|532924;152422|B
11:47:37,999|REFRESH|532924;152422|B
11:47:38,440|SYNCFAILED|532924;152422|B
```

Sample server logs

```
11:57:31,367|532958;152588|C|stroke|
11:57:31,491|533281;152516|A|refresh|
11:57:32,462|532960;152595|C|refresh|
11:57:35,100|533281;152516|A|refresh|
11:57:35,955|532960;152595|C|stroke|
11:57:36,028|532960;152595|C|stroke|
11:57:36,730|532960;152595|C|position|
11:57:38,627|532960;152595|C|stroke|
```

Positioning

One of the pre-requisite to do an indoor positioning system for CatchBob! was that no extra infrastructure had to be setup. We had to use what was available. GPS (Global Positioning System) was out of question because it requires a line of sight to the sky therefore does not work indoors (and neither in dense urban environments). There are basically four ways to do indoor positioning: with infrared beacons, radio beacons, ultrasound systems and video-based systems. These technologies can be combined or augmented with outdoor positioning systems like GPS to extend the coverage or for calibration. Radio beacon based positioning with Wi-Fi was the only alternative. When I started to fool around with Wi-Fi positioning, the EPFL campus had around 100 access points. Some buildings and open areas where not covered at all. After some politics, the logistic and infrastructure department agreed to provide me with a well updated list of all the Wi-Fi antennas. The list contains the access point's MAC address and the label of the room where it is located.

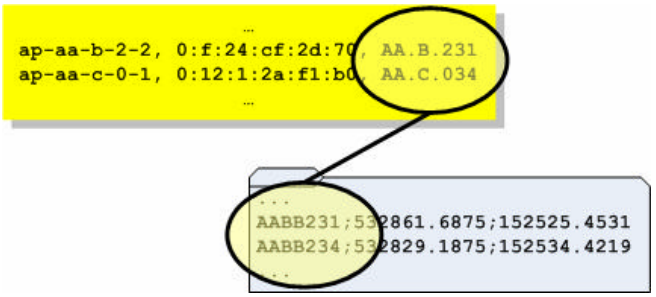


Figure 6: Mapping the labels. Matching the data provided by the network people and the infrastructures people. An access point has a label, a MAC address, and a room. A room has a label and inside x and y coordinates.

The team providing the map service for our campus nicely provided me with a dump of there database containing rooms label and their x,y positions in Swiss coordinates and their pseudo-z coordinate (i.e. their level in the building). Unfortunately they do not have the same naming convention for labeling the room in the same way as the

EXPERTS SAY

Indoor positioning technologies relate people, objects and events in space. Building a positioning system that works well indoors is a challenge, because signals reflected of walls, floors and ceilings tend to confuse sensors, and often there are obstructions between sensors and objects being tracked.

Kris Kolodziej [41]

EXPERT SAY

Building and deploying location-aware application that are usable by a wide variety of people in everyday situations is arguably no easier than it was ten years ago

Yu-Chung Chen, UCSD, 2005

logistic and infrastructure department people.

I had to write a label converter to mix the data that I run whenever the access point table is updated. I keep all these data in a database that I use to generate a flat text file that I use for my Wi-Fi positioning application including CatchBob! The file looks like this [AP MAC address|X|Y]:

```
001201396e50|532798|152528
0012012af1b0|532813|152557
000785b35d33|532845|152555
```

Currently, there are more than 200 access points spread all over the campus. However, a burst in the number of access points does not necessarily means better accuracy, better coverage or less cold spots. Simply because it is more important to densify the areas where the mobile users stay by doubling or tripling the number of access points. These areas include large auditoriums, popular cafeterias, libraries, and mini learning centers. They become the campus's hotspots. Buildings with few administrative offices are not a priority because less prone to host mobility. More Wi-Fi beacons means many times more density on specific areas than improved overall coverage.

[positioning techniques]

There are two main solutions used in WLAN positioning. The first one, which is called the empirical model, is based on storing pre-recorded measurements in a database. Before any location positioning can be performed, a radio map is created. This map stores the signal strengths at each particular location from all access points that can be read from that area into a database. When a device requests a location, it matches signal strengths from all the access points that it can read with the database.

The second solution is called the propagation model. It is based on the degradation of the signal strength of a radio wave over distance in space. The accuracy of this model is dependent on the algorithm that is used.

CatchBob! is based on the propagation model. Mainly because the EPFL WLAN network is constantly changing and expending. Constantly updating and calibrating the radio map would have been too fastidious. Besides the high positioning accuracy was not a primer goal. CatchBob! uses triangulation to compute the positions of the players. Nowadays, triangulation is used for any kind of geometric approaches for location. Originally, a triangulation needed two access point positions. From each position we measure the angle to the location. Geometrically speaking, we get the location if we intersect two lines.

Ekahau [19], THE Wi-Fi positioning company claims to reach an accuracy of 1-2 meters, RADAR [20] and Cricket

EXPERT SAY

An important tradeoff while deploying such a wide area location system is the accuracy of the positioning infrastructure versus the calibration effort involved.

Yu-Chung Cheng, UCSD

JARGON WATCH

*A **propagation model** is based on the degradation of the signal strength of a radio wave over distance in space.*

*The **empirical model** is based on storing pre-recorded measurements in a database. A so-called "radio map" is constructed before location positioning can begin. The radio map is the site map that contains markings of a series of selected points [42].*

JARGON WATCH

*Nowadays, **triangulation** is used for any kind of geometric approaches for location. Originally, a triangulation needed two access point positions. From each position we measure the angle to the location. Geometrically speaking, we get the location if we intersect two lines*

[21] can provide accurate estimate within 2-4 meters. However, this precision is achieved on tightly controlled conditions, with many hours of installation and calibration. I found rather early in the project that implementing the right positioning algorithm for the topology of the CatchBob! playground was a daunting task and a research field in itself. Moreover, a rough positioning accuracy of 5-20 meters appeared to be enough to achieve the collaborative task of CatchBob! Therefore, I only implemented a very simple centroid algorithm. I position the user at the center of the scanned nearby access points by computing an average of their x, y location. In addition I weighted by the received signal strength during a scan. Other positioning algorithms [22] include Fingerprinting and Particle Filters.

[NIC sniffing]

CatchBob! was originally developed for and experimented on iPAQs 5550. Therefore I developed CraftDeamon. CraftDeamon is a C++ DLL which calls NDIS functions on the IEEE 802.11 NIC and returns the detected BSSIDs and their attributes stored in the IEEE 802.11 NIC's database. CraftDeamon's methods are exported to be invoked by C# clients. To test it, I developed CRAFTStumbler which displays the access points' MAC addresses and signal strength and if selected redirects the data to a remote server which logs them on a centralized database. It is suggested that Wi-Fi scanning based on NDIS functions should not be done at a lower rate than every 6 seconds.

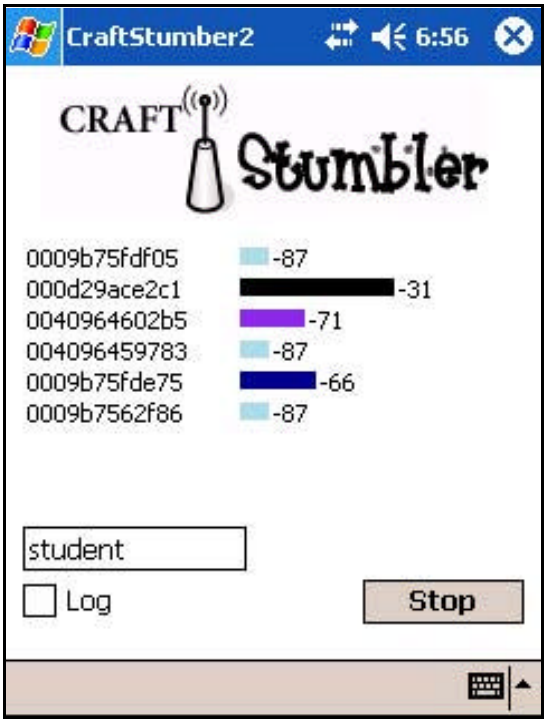


Figure 7: CRAFTStumbler, a very simple war-driving tool I developed for PocketPC. It lists and logs every 6 seconds the MAC addresses and the signal strength of the nearby Wi-Fi

JARGON WATCH

*The **centroid algorithm** computes a location with the arithmetic mean of the access points position weighted by their signal strength.*

JARGON WATCH

*Network Driver Interface Specification (**NDIS**) is a standard API of LAN drivers for "Network Interface Cards" (NIC's) developed by Microsoft*

JARGON WATCH

*The **BSSID** is the MAC address of the access point (e.g. 00:de:ad:be:ef:00)*

JARGON WATCH

*In **active scanning**, the client sending probes out on all its channels looking for access points.*

*In **passive scanning**, the client relies on network traffic to discover access points.*

access points.

A problem with the active scanning that I used is that access points can sometimes be configured to never send out broadcast beacon packets announcing their presence either. In such scenario, passive scanning where the Wi-Fi card does not send out probe requests, and instead simply sniffs traffic on each of the Wi-Fi channels, may be used.

[tracking]

Based on the CRAFTStumbler logs I generated my first map in SVG. I tracked a co-worker doing a 25 minutes walk on the campus. I used a map knowing its x,y origin in Swiss coordinates as well as the x and y value of the bottom right corner of the map.

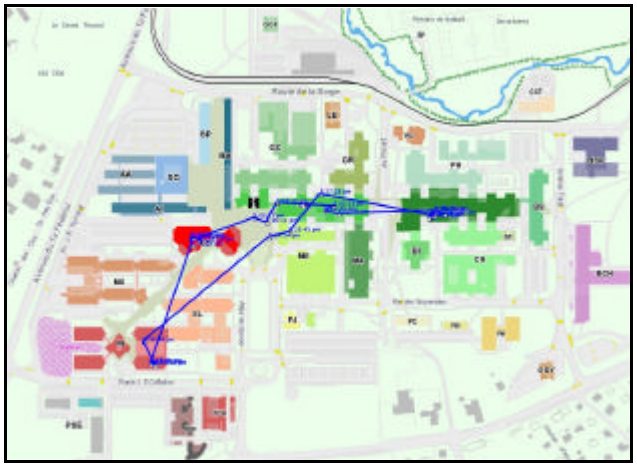


Figure 8: Tracking a colleague

Back in the days (end of 2003) the only way to do Wi-Fi sniffing on the iPAQ5550 was to apply the Australian restaurant principle: BYO (well, not Bring you own, but rather Build you own) and not many people were willing to share their code. Nowadays, there are frameworks like Place Lab [23] or the Mobile Bristol Toolkit [24] that do the work perfectly. When we decided to move CatchBob! to TabletPC, I immediately picked Place Lab to handle the NIC card sniffing. Place Lab is cross-platform (OS specific libraries exported to be used in Java) and is compatible with the many types of hardware. It runs smoothly except that I think it relies on Microsoft's Wireless Zero Config (WZC). I believe this will be improved.

Communication

Any network programming is about dealing with availability, latency and reliability... and even more so with mobile devices.

I started to write a specific protocol for CatchBob! for its PocketPC version. A standalone Java server with a C# client. It was rather challenging to write sockets that can tolerate the unreliability of the connectivity of a mobile device. Especially, when the .NET compact framework

JARGON WATCH

*Scalable Vector Graphics (**SVG**) is an XML markup language for describing two-dimensional vector graphics, both static and animated*

EXPERTS SAY

We are forced to consider hybrid approaches, which combine the intermittent connections and latency tolerance of turn-based games with the gameplay of action titles.

Eben Upton [43]

was not properly handling the timeout on sockets. Nerve-racking. Migrating CatchBob! to TabletPC, made me change my strategy to handle the client-server communication. First I was happy to go back to Java (it was easier to develop with C# on iPAQ than in Java). I had to extend the protocol to handle stylus stroke synchronization and was expecting the protocol to change quite a few times until we really knew what we wanted to do with our TabletPC client. Interoperability was not to be forgotten either. I also had to deal with a strengthen security on the university network with a restricted access of all ports except the 80. All these factors mixed naturally lead me to SOAP (Simple Object Access Protocol) and Axis (the Apache implementation of SOAP). Even running a beta version of Axis 1.2, I could simplify the development, build and deployment process of both the server and client. For this, I had to pay the price of now over-sized XML messages. SOAP can easily become bandwidth hungry [25]. It has not been too much of an issue for CatchBob! because there is no need to scale-up to a massive multi-player game or have a mission critical type of latency. Actually, I thought that mobile users, in general, were used to bad latency and I must say that improving was not my focus. I misjudged here the fact that some players did not categorize a TabletPC in the same mobile client category as a PDA or a cell phone. Bigger screen meant bigger expectations on positioning and accuracy and network availability.

One trend in location-based mobile games is to play with the tolerance and reliability of the technologies used. Positioning errors, maps approximations and network unreliability become part of the games. In CatchBob! we question if sharp positioning accuracy improves the collective performance on the task. Other groups come with the concept of seamfulness. A seamful game [26] is a GPS and Wi-Fi based game that harness negative aspects of infrastructure technologies, which are normally concealed and unexplained, and present them as game features allowing users to explore and understand them.

User Interface

Let's put it this way, back in my engineering school days, taking the Human-Computer Interaction course was for students in desperate need for credits. CatchBob! made me reflect on a more user-centered design. Thinking about a scenario of the usage. A user's need is not a system functionality. Put that in your pipe and smoke it!

[channels of communication]

We first designed a CatchBob! on PocketPC with a simple type of communication. Drawing a line from a player to a position was is only way to interact between the players. The interaction was too poor. For example it was difficult to express one's strategy by just drawing recommended directions. We decided it was better to move bigger and

JARGON WATCH

*A wireless LAN coverage within an area goes "cold" and creates **cold spots** because walls get in the way, access points are not well-placed or other reasons.*

*A **ruban canyon** is formed by building preventing GPS units from seeing enough satellites to get a position lock.*

JARGON WATCH

***Seamful games** are games that deliberately but selectively revealing these the edges and gaps in wi-fi cells to users, to let people make of them what they will.*

increase the communication channel. We discussed to add VoIP (difficult to record and transcript 3 players) or a chat or a semi-structured chat (too poor also). We finally thought that freely and synchronously annotating the map was the way to go.

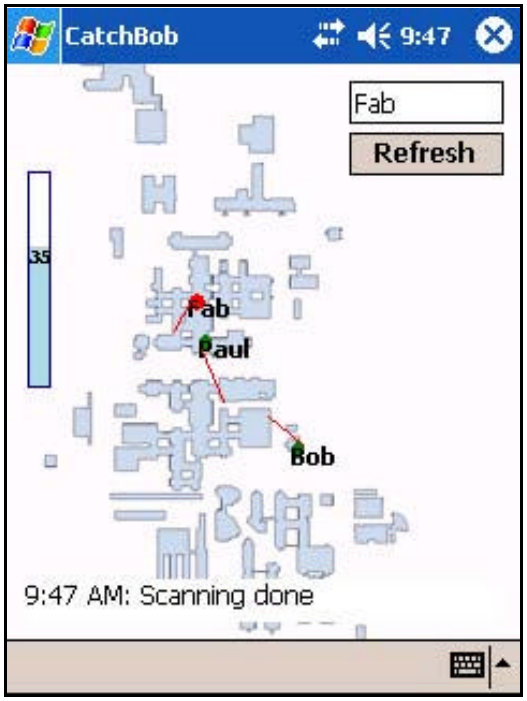


Figure 9: CatchBob! interface on PocketPC. Players are displayed as dots and the red line are the move recommendation they give to each other. The proximity sensor is on the left.

[stylus strokes]

Capturing the stylus drawings was easy. However, compression was needed to avoid sending orca-fat synchronization messages in SOAP. I found out that only recording one pixel out of 4 was good enough.

CatchBob! has no eraser. The annotations constantly fades out. After 4 minutes, an annotation is not visible anymore. 4 minutes was chosen arbitrarily and proved to be the right timing for annotation to become outdated.

I try to give the perception of annotation synchronization to the user. Local (not synchronized yet) annotations are drawn in black and the shared (synchronized) annotations are displayed with the player's color. Stylus strokes are shared via the server as soon as the interface detects that the player is not drawing anymore.

[users perceptions]

There was a lot to learn listening the CatchBob! participants talking about the application and the interface. There was a gap between the user's expectations and what the tool really does. Sometimes it lead to the lost of trust on the tool "it did not act like I thought it would have" and frustration "sometimes it

Figure 12: Historic of the TabletPC interface

refused to work". An unreliable network is sometimes not perceived as a disturbance but as a failure. A couple of users even blamed the tool for the lost of connectivity. It is difficult to make understand the state of connectivity. Mainly because it is hard to detect a break of connection. A client can still detect nearby Wi-Fi access points, while not having enough signal to send and receive data. In that case, displaying the signal strength might not be enough.

The positioning accuracy was not always understood and disturbed some players "I did not move physically, but I moved on the map" and "The proximity to Bob changed even though I did not move". This was rather a surprise. Players came with a pre-conception on the quality of indoor positioning systems. It is still unclear to me why. GPS systems have an accuracy of 5-10 meters. Neither is it clear how bad and good positioning accuracy could be displayed.

Users were not used to write on a TabletPC. Some had a hard time adapting their writing, some other tried to challenge the tool by writing as fast and small as possible.

Funnily enough, a player got lost for a few minutes while playing. He argued that he could not tell where the north was on the map. I automatically added a compass card on the top right of the map. When hold properly, the TabletPC displays the map with the north at the top. Apparently, the user was not holding the TabletPC properly and therefore did not interpret the map correctly.

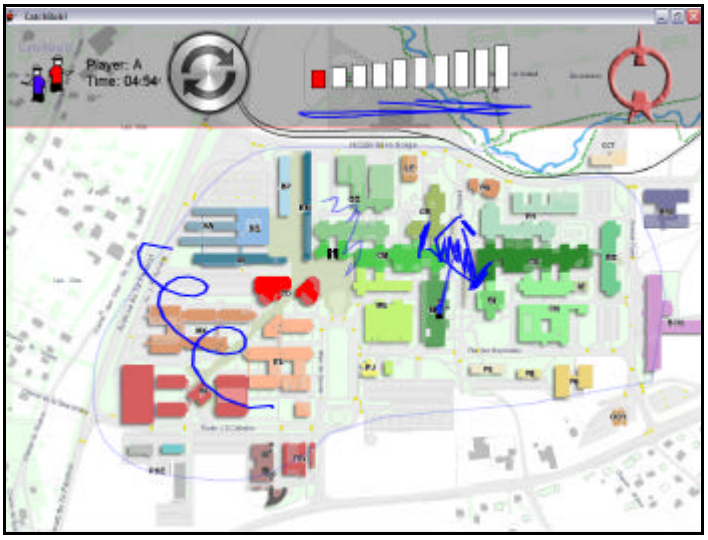


Figure 10: CatchBob! interface on TabletPC

[replay tool]

After playing the game, in order to gather more information about how they used the tool participants are self-confronted with a replay of their paths on the campus. We ask participants why they found the interaction

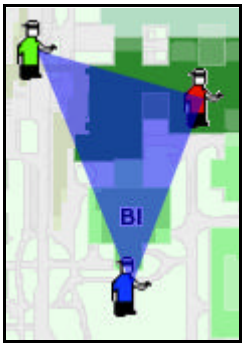


Figure 13: Players are forming a triangle to surround Bob and catch him.

episode interesting, why they did that. In order to do so, we designed a replay tool which provides an interface to navigate through the game, and show the history of collaborative episodes/interactions. To replay, data are taken from the server logs.

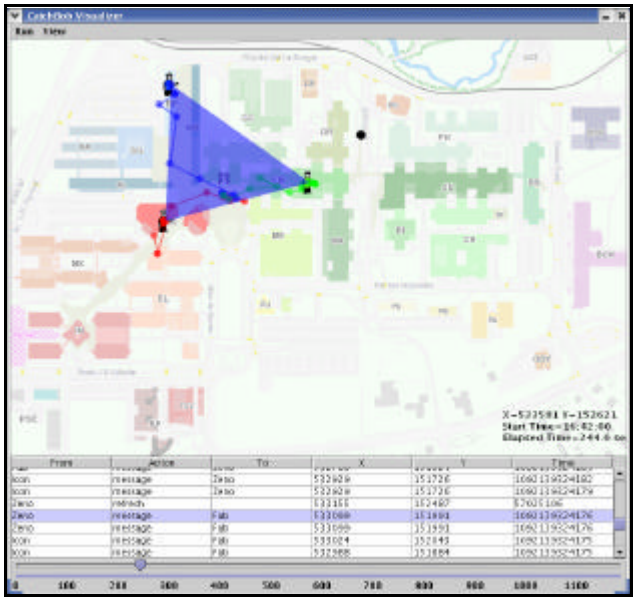


Figure 11: Players are self-confronted with a replay of their paths on the campus taken from the server log

[introducing the game]

When we started the experiments, we were introducing the CatchBob! interface with a small tutorial and screenshots. We found out early that it was not enough. Some players were not grasping the meaning of the refresh button or would have troubles using the stylus. The players need to touch the tool before they use it. Otherwise, they would spend too much time to own it or they would purely misunderstand it. We also failed at first to transfer trust in the tool to the players. I was mentioning "be gentle with it", but I quickly learned to shut up, transmitting the TabletPCs with confidence and as if nothing bad ever happened in the pre-test phase.

Discussion

Locative media is at the intersection of the virtual and physical spaces. They are information systems with geographical data that allow the numerical and the physical worlds to mesh and converge.

As we could see in CatchBob!, geolocation is still a very disturbing technology that is not yet part of our habits. Moreover, most of the current experiments are technology oriented and I do not take into consideration the human factor, the users and their activities. Nicolas's research on spatial awareness is one step in that direction.

While building locative and collaborative application like

JARGON WATCH

ShoutSpace *is a geographical messaging system for the EPFL campus community as a segway to an ontology of spatial communication*

CatchBob! and now ShoutSpace, I touched many different topics (from deep down technical to very user oriented). Some I find relevant to introduce here.

[from position to place]

CatchBob! does not carry the sense of place. However it would be interesting to change the players' position coordinates to a building name or a self-labeled space. Moreover, how useful is the positioning accuracy and when is it important is it to know a position rather than a place? People do not infer the same when it comes to a system based on position or place. The notion of place would also be interested to support the technology with human self-positioning. For example to calibrate and refine a positioning system.

Following is a rough summary of "From Position to Place" [27] by Jeffrey Hightower: The need for application to reason about "place", not coordinates. Existing systems rely on manually defining places which, while useful, does not scale to ubiquitous deployment. Generically, place is a human-readable labeling of positions. Current approaches require manual definition of places. I must, by hand, delineate and label my neighborhood, property, rooms, furniture, and service areas of my devices. Manual definition does not scale. Instead, ubiquitous deployment requires automatically learning significant regions and semantically labeling them as places.

The challenge is to augment maps of physical features with the dynamic data to, over time, suggest geometric regions which are good candidates to label as places. Label directly represents the place's demographic, environmental, historic, personal, or commercial significance and is the desired abstraction for emerging proactive applications.

The interface should be capable of answering both "What place labels are associated with my current coordinates?" and "What is the probability I am currently is in a place P?"

[3D Wi-Fi positioning]

Very early in the CatchBob! project I was asked if it was possible to do 3D Wi-Fi positioning with the infrastructure we had on the campus. I always answered with the negative. The problem is that the 3rd dimension, the altitude component of location, is a discrete data. Of course, technically, it is possible, by mapping the beacons to the level in a building. It is actually properly done at the EPFL, with Wi-Fi beacons labeling convention "ap-[building acronym]-[level]-[AP number]" (eg "ap-ai-1-2"). I could have taken the third component of the label and calculate an approximate level. However, since my positioning algorithm was already giving approximate 2D, I did not want to add another factor of approximation. I doubt the players would have understood the 3D

EXPERT SAY

ubiquitous deployment requires automatically learning significant regions and semantically labeling them as places.

Jeffrey Hightower

EXPERT SAY

Our current belief is that generating "2.5" dimension estimates in which altitude is represented with a symbolic name are more meaningful than a coordinate-based altitude.

Anthony LaMarca

positioning failure, especially when we ask them to trust the system. Nevertheless, 3D Wi-Fi positioning in a seamful game would be of a higher interest.

People at the Intel Research Center in Seattle have a 2,5D positioning approach. They introduce it in "Place Lab: Device Positioning Using Radio Beacons in the Wild" [28]: For many emerging location-aware applications it is much easier to utilize place names like "Bank", "Starbucks" or "Movie Theater", than geo-coordinates such as (48.43456, -122.45678). We hope to develop techniques that allow Place Lab to automatically learn and estimate actual place names in addition to geo-coordinates. A first step in this process is moving Place Lab to "2.5" dimensions. Place Lab currently only generates position estimates in two dimensions (latitude and longitude) and ignores the altitude component of location. This can present a problem in multistory buildings where floor number is likely a key aspect of location. Our current belief is that generating "2.5" dimension estimates in which altitude is represented with a symbolic name such as "Parking Level A" or "3rd floor" are more meaningful than a coordinate-based altitude like 34.6 meters above sea level. We are planning to augment Place Lab to allow beacons and traces to be annotated with floor information and have our trackers predict this symbolic dimension along with latitude and longitude.

[spatial annotation]

CatchBob uses synchronous spatial annotation. Other locative projects are based on asynchronous spatial annotation [29] [30] [31]. The use of spatial annotation is clear for a much defined coordinative task like the one in CatchBob! Things are less evident for asynchronous and linear annotation of space. What makes most of the projects fail or why do not they attract many participants? There is plentiful of literature on online communities building. They talk about social status, community of practice and the natural formation of hierarchy (from newbies to master). Space brings two more variables to the equation. Diversity and density. The space needs to be diverse to engage annotation. A remote campus is not as rich in diversity as London city center. The annotating community needs to be dense in terms of interests, community of practices need to be formed.

People at the Institute of the Future in Palo Alto lead a workshop on Why people will geo-annotate physical space? [32] For them the simplicity of the interface was crucial. "What's most crucial for this future to thrive? Interface. Simple, convenient interfaces for inputting annotations and uncluttered, filtered interfaces for receiving annotations. Trust systems, social networks, and personal profiles & preferences will be necessary tools for survival when every urban street corner may have hundreds of annotations."

EXPERT SAY

Most current location systems do not work where people spend much of their time

Yu-Chung Cheng, UCSD

Some other argue that annotation must foster conversation, because tagging and annotation are sterile. It does not matter if the topic of the conversation fades, as long as the conversation goes on. Urban planners think about spatial annotation to build a conditional public trust that makes public places livable [33].

[engaging technology]

Players mentioned in their post-interview that they genuinely enjoyed CatchBob! Now what if it was not a part of an experiment in an institute of technology? The game scenario put aside, would the technology we used and the way we interfaced it be embraced by a majority of people? This question is the key to any user-centric technology, but even more especially so in the "everyware" world, because ubiquitous, pervasive and geolocated systems are about to enter into people's lives. They will tend to become inevitable, but it does not mean they will be accepted. Moreover, there is a danger that these technologies present users with unacceptable difficulty, confusion and uncertainty.

For the locative media to thrive there is a need for engaging technology. There is a need to create a technology that can be embraced by a majority of people. This might not necessarily be the job of engineers because they already given a narrow technical brief and constraints and it is not their mandate to consider the social environmental impact of their work.

Adam Greenfield, came up with a detailed and thoughtful essay on designing ubiquitous computing systems [34]. He provides a set of principals, a couple of them being very relevant to locative media in my sense: The systems must offer users the ability to opt out, always and at any point, they must not introduce undue complications into ordinary operations.

Conclusion

Building a mobile, locative, and collaborative application is a challenging engineering task because it touches many aspects of software computing, from networking to the fields of human-computer interaction, and social computing like carrying the sense of connectivity or providing a social context.

Current technologies have their imperfections and we have to play around them or mix them. The limits are now known. Engineers have been doing their work by providing frameworks and toolkit for Wi-Fi positioning and for data communication. Building location-based application is being simplified. CatchBob! is a clear example of these achievements. Nevertheless, I lot still needs to be done for locative media to reach maturity, to strengthen the link semantic between the numerical and physical world. No

Expert Say

If ubicomp applications are rushed to market and allowed to appear as have so many technological artifacts in the last thirty years, then they will present those users with a truly unprecedented level of badness

Adam Greenfield

JARGON WATCH

***Everyware** is the name for the new class of software needed to power the world of ubiquitous computing.*

matter how good the technology is, it has to be useful and useable, it has to be engaging!

Acknowledgements

Many thanks go to Nicolas Nova for his extremely valuable insights, work, and not-too-fuzzy requirements. Patrick Jermann for his hacks on collecting and playing with the maps and the data. The proofreaders of this document, namely Pascal Betz, Mauro Cherubini. Prof. Pierre Dillenbourg for trusting creativity

About the author

Fabien Girardin is a software engineer at the Swiss Federal Institute of Technology in Lausanne (EPFL), member of the CRAFT, a learning technology lab, with a focus on technologies for collaboration. He has previous experience in decentralized and distributed systems. His current interests are in mobile computing, computer supported collaborative work and locative media. One of his ongoing projects is the development of ShoutSpace [35], a spatial annotation system at the EPFL.

Email: Fabien.Girardin@epfl.ch
WWWeb: http://people.epfl.ch/Fabien.Girardin

---

1 CatchBob! (2004) http://craftsrv1.epfl.ch/research/catchbob/

2 Center for Research and Support of Training and its Technologies (CRAFT) http://craft.epfl.ch

3 PLAN, pervasive & locative arts network (2005) http://www.open-plan.org/

4 Uncle Roy All Around You (2002) http://www.uncleroyallaroundyou.co.uk/

5 Can You See Me Now? (2003) http://www.blasttheory.co.uk/bt/work_cysmn.html

6 Location-based mobile phone games (2004) http://www.in-duce.net/archives/locationbased_mobile_phone_games.php

7Nicolas Nova, Fabien Girardin (2004) Analysis of a Location-Based Multi-Player Game. Games and Social Network: a Workshop on Multiplayer Games, British HCI conference

8 Joel Spolsky (2004) Joel on Software: And on Diverse and Occasionally Related Matters That Will Prove of Interest to Software Developers, Designers, and Managers, and to Those Who, Whether by Good Fortune or Ill Luck, Work with Them in Some Capacity. Apress

9 Games and Social Network: a Workshop on Multiplayer Games (2004) http://www.dcs.gla.ac.uk/~barry/gamesworkshop/

10 Nicolas Nova Running Notes at the Games and Social Networks in Leeds (2004 http://tecfa.unige.ch/~nova/leeds06092004.txt

11 Schuyler Erle, Rich Gibson, Jo Walsh (2005) Mapping Hacks, O'Reilly

12 WarDriving (2004) http://www.wardriving.com/

13 FreeNetworks.org (2004) http://www.freenetworks.org/

14 Jo Walsh, http://space.frot.org/

15 Wireless London, http://wirelesslondon.info/

16 LondonFreeMap (2004) http://uo.space.frot.org/?LondonFreeMap

17 Use Wi-Fi To Play Access Point Games, http://www.extremetech.com/article2/0%2C1558%2C1746261%2C00.asp

18 Pascal Betz (2004) Hibernate - XDoclet Tutorial, http://www.downside.ch/hibernate/

19 Ekahau (2003) http://www.ekahau.com/

20 P. Bahl and V. N. Padmanabhan (2000) RADAR: An In-Building RF-Based User Location and Tracking System. In Proceedings of IEEE Infocom 00.

21 N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The Cricket Location-Support System. In Proceedings of ACM MobiCom'00, July 2000.

22 Yu-Chung Cheng, Yatin Chawathe, Anthony LaMarca and John Krumm (2005) Accuracy Characterization for Metropolitan-scale Wi-Fi Localization. to appear in Proceedings of Mobisys 2005

23 Place Lab (2004) http://www.placelab.org/

24 Mobile Bristol (2004) http://www.mobilebristol.co.uk/

25 Forum Nokia (2003). Optimizing the Client/Server Communication for Mobile Applications, Part 3

26 Seamful Games (2005) http://www.seamful.com/

27 Jeffrey Hightower (2003) From Position to Place

28 Anthony LaMarca, Yatin Chawathe, Sunny Consolvo, Jeffrey Hightower, Ian Smith, James Scott, Tim Sohn, James Howard, Jeff Hughes, Fred Potter, Jason Tabert, Pauline Powledge, Gaetano Borriello and Bill Schili. (2004) Place Lab: Device Positioning Using Radio Beacons in the Wild. Pervasive 2005, Munich Germany

29 System for TAgging Messages, Post-Inferential Semantics (STAMPS) (2005) http://craftsrv1.epfl.ch/research/stamps/

30 Urban Tapestries (2003) http://urbantapestries.net/

31 GeoNotes (2002) http://geonotes.sics.se/

32 Why people will geo-annotate physical space (2004) http://blogger.iftf.org/Future/000599.html

33 Townsend A M. (2003). "Wired/Unwired: The Urban Geography of Digital Networks" Unpublished doctoral dissertation. Massachusetts Institute of Technology.

34 All watched over by machines of loving grace: Some ethical guidelines for user experience in ubiquitous-computing settings (2004) http://www.boxesandarrows.com/archives/all_watched_over_by_machines_of_loving_grace_some_ethical_guidelines_for_user_experience_in_ubiquitouscomputing_settings_1.php

35 ShoutSpace (2004) http://craftsrv1.epfl.ch/research/shoutspace/

36 Pasta & Vinegar (2003) http://tecfa.unige.ch/perso/staf/nova/blog/

37 Place Lab (2004) http://www.placelab.org/

38 Mobile Bristol (2004) http://www.mobilebristol.co.uk/

39 SOAP Specifications (2002) http://www.w3.org/TR/soap/

40 P.T. Eugster, R. Guerraoui, A.-M. Kermarrec, L. Massoulié (2004) From Epidemics to Distributed Computing. IEEE Computer, 2004

41 Kris Kolodziej (2004) Indoor Location Technology Opens New Worlds
http://www.geoplace.com/gw/2004/0406/0406lt.asp

42 Johnny Shih (2003) Wireless LAN Location System,
http://innovexpo.itee.uq.edu.au/2003/exhibits/s358272/thesis.pdf

43 Eben Upton(2004) Realtime Multiplayer Games over Mobile Networks,
http://www.dcs.gla.ac.uk/equator/gamesworkshop/papers/upton.pdf