Place Lab: Wide-scale Device Positioning Using Radio Beacons in the Wild

Anthony LaMarca[1], Yatin Chawathe[1], Ian Smith[1], Sunny Consolvo[1], Jeff Hightower[1], James Scott[2], Tim Sohn[4], Pauline Powledge[1], Anand Balachandran[3], Gaetano Borriello[5], Bill Schilit[1]

[1]Intel Research Seattle
[2]Intel Research Cambridge
[3]Microsoft Corporation
[4]Department of Computer Science, UC San Diego
[5]Department of Computer Science & Engineering, University of Washington

Place Lab is a location system that estimates a user's position by scanning for and triangulating the position of known radio sources in the environment. Systems that compute the position of users and devices in the physical world have long been a key component in pervasive computing deployments. This stems from the critical role that context plays in pervasive computing applications and the importance of location to a mobile user's context. Given this strong motivation, dozens of research and commercial location-tracking systems have been built using technologies ranging from ultrasonic time-of-flight, infrared proximity, radio frequency signal strength and time-of-flight, and electro-magnetic field strength. Using these systems many location-aware applications and services have been deployed spanning a variety of application domains[1].

Despite these successes, location-aware computing is stuck in an unfortunate cycle. Nearly all location-tracking systems require expensive infrastructure and/or a large amount of calibration. Since this is generally not a barrier to those of us in research, it has not kept us from innovating in the application space. It has however, drastically hindered the adoption of these application by real users. The result is that while we can give compelling demonstrations of location-based applications, few can be used in the places they are most useful: where we live, where we socialize, where we shop. This create the following unfortunate cycle: there are very few users due to the dearth of applications for them to run; developers are not interested in writing applications that require absent infrastructure; infrastructure investments are based on user demand, which there is little of at this time.

To attempt to break this cycle, we have developed Place Lab, a location system with an extremely low barrier to entry. Place Lab uses the device's wireless networking interfaces to listen for nearby radio sources like 802.11 access points, other Bluetooth devices and GSM towers. These technologies and others assign unique or semi-unique IDs to the radios sources, and these IDs can be scanned for by clients. Devices running Place Lab consult a local cache of known radio sources and use these observed IDs to look up the location of the radio sources. By triangulating the location of these sources, devices can estimate their location. The coverage of Place Lab is based on the availability of radio sources. Fortunately wireless networking infrastructure, especially 802.11, is being deployed at a tremendous pace. Many places have complete GSM coverage and cities such as Westminster (a London borough) are expected to be fully covered with 802.11 access points in the near future as well. Our own measurements in downtown Seattle showed an 802.11b density of 1200 access points (APs) per km^2. The accuracy of Place Lab is based on both the number and type of signal that is observed. Hearing more sources allows a more accurate estimate to be generated, as do signals from short range sources, for example Bluetooth. As a sample, a Place Lab client hearing 3 or 4 knows 802.11 access points will typically predict its location within 20 meters of the truth. By utilizing a local cache of known radio sources, a Place Lab client using a technology that allows passive scanning (like 802.11) position itself completely locally, with no transmission or communication with the infrastructure. This gives the user complete control over when their location is disclosed, laying the foundation for privacy-observant location-based applications.

Place Lab runs on a wide variety of platforms including both cell phones and more capable devices like laptops and PDAs with wireless networking. Place Lab does not rely on custom infrastructure and instead makes use of the existing radio sources in the user's

[1] Lest there be any lingering thoughts that location-aware computing is still a research oddity, the reader should note that the next major version of the dominant consumer operating system, Microsoft Windows Longhorn, is slated to include a "Location API" that will provide a uniform interface for applications to access the physical location of the user's device.

UIST '04, October 24–27, 2004, Santa Fe, New Mexico, USA

ACM 1-58113-962-4/04/0010

environment. Place Lab does rely on a database of mapped radio sources. To increase coverage, Place Lab clients can fill their cache from a variety of user-contributed radio databases, including data collected by war drivers, WiFi clubs and radio enthusiasts. As of July 2004, Place Lab clients had access to data sets containing 1.6 million mapped radio sources. To encourage its adoption, Place Lab has been released under an open source license. Binary and source releases as well as sample radio traces can be downloaded from http://www.placelab.org.

With the exception of the small amount of native code that is written for each spotter, Place Lab is written entirely in Java. To achieve a high degree of portability, Place Lab is implemented in Java 2 Micro Edition (J2ME). Place Lab currently runs on the following platforms and provides support for spotting the following beacon types. In addition to the listed types of beacons, all platforms support serial and Bluetooth GPS devices.

Operating System	**Architecture**	**Beacon Types**
Windows XP	x86	802.11abg, Bluetooth
Linux	x86, ARM, Xscale	802.11abg
OS X	Power PC	802.11abg, Bluetooth
Pocket PC 2003	ARM, XScale	802.11abg, Bluetooth
Symbian	Nokia 40 and 60 series	GSM, Bluetooth

Applications that conform to the core Place Lab interfaces can be written in a platform and beacon independent way. That is to say, carefully written Place Lab applications can be moved from Windows to a Macintosh to a cell phone without requiring any rewriting or recompilation.

We do not wish to suggest that making such portable applications is easy. The cell phones do not support UI toolkits like Swing or SWT and in some cases (such as the Nokias we support) do not even support floating point. Thus application wishing to be extremely portable must take care to use the fixed point location interfaces as well as the limited UI components available to the phone's Java profile. We hope that the wide availability and applicability of Place Lab's device positioning technology offers the research community a new tool to explore location enhanced computing in real—in the wild—environments.