Practical Lessons from Place Lab

Jeffrey Hightower and Anthony LaMarca and Ian E. Smith
Intel Research Seattle
1100 NE 45th St. 6th Floor
Seattle, WA 98105

September 2006

Practical Lessons from Place Lab

The creators of Place Lab, a location-enhanced mobile computing system, share lessons learned about leading the project and real-world deployment.

Jeffrey Hightower, Anthony LaMarca, and Ian E. Smith
Intel Research Seattle

In pervasive computing devices, a device's ability to discover its location often proves useful. To achieve this, many research and commercial location systems have used technologies such as radio signal strength, radio time-of-flight, ultrasonic time-of-flight, infrared proximity, and optical vision. The most successful system, the Global Positioning System, is satellite-based, so it works worldwide. Inexpensive GPS receiver chips are also easy to build into devices. However, because GPS receivers require a clear view of the sky, they typically don't work indoors, under cover, or in many urban canyons. When we started our work in 2003, most location systems other than GPS relied on infrastructure that limited coverage to a room or building. Inexpensive ubiquitous positioning systems didn't exist, slowing the widespread adoption of location-enhanced computing. We built Place Lab to address the need for a ubiquitous location capability.

The project

Place Lab lets a device estimate its location by scanning for fixed radio beacons—such as nearby 802.11 access points and GSM (Global System for Mobile Communications) cell towers—and referencing the beacons' positions cached in the mobile device's database. To enable real-world deployment, we set these goals for Place Lab:

- *Work over a wide area, indoors and out.* We chose 802.11 and GSM signals, which are being rapidly and ubiquitously deployed throughout populated areas.
- *Run on commodity devices.* Place Lab uses everyday device hardware and whatever radios a device is equipped with. It can run on laptops, PDAs, and cell phones with different architectures and operating systems.
- *Observe privacy needs.* Users' devices estimate their locations locally, without requiring a network connection or any server-based infrastructure.
- *Support standard programming interfaces.* To support existing applications, Place Lab exposes its location estimates in industry-standard APIs, including NMEA 0183 (the National Marine Electronic Association protocol used by GPS receivers) and Java Specification Request 179 (see www.jcp.org/en/jsr/detail?id=179).
- *Make accuracy a secondary goal.* Unlike previous location research efforts that focused on maximizing the location estimate's accuracy, we were willing to trade some accuracy for ubiquity. Similarly, mobile phones became invaluable tools, not because of their high voice quality, but because they work in most places, with good-enough voice quality.

To achieve these goals, we developed a location system and supporting applications comprising approximately 100,000 lines of code. We performed a *wardrive* (a search for Wi-Fi networks by moving vehicle) of more than 4,350 kilometers to construct a database of the locations of approximately 35,000 Wi-Fi and 7,000 GSM beacons in the Seattle metropoli-

Published by the IEEE CS and IEEE ComSoc ■ 1536-1268/06/$20.00 © 2006 IEEE

tan area. Place Lab devices can access the wigle.net community Wi-Fi database, which increases coverage to almost six million beacons worldwide. As of April 2006, more than 16,000 unique IP addresses have downloaded the Place Lab toolkit (http://placelab.org).

Our technology evaluations and user studies

We conducted nine user studies with a total of 88 participants. We also did several empirical technology evaluations.

The basic idea behind Place Lab is *beacon location*, a viable approach to achieve ubiquitous location capability.[1] We evaluated 802.11 access point density and quantified the relationship between beacon density and location accuracy in three representative neighborhoods of the greater Seattle area—urban, residential, and suburban. Our experiments showed that if beacon density is high enough for devices to see at least three distinct beacons during a 10-second window, Place Lab can achieve a median location error of 15 to 20 meters. Beacon location is less accurate than GPS but, unlike GPS, covers nearly 100 percent of users' locations. In the sparsely populated suburban area where 802.11 density is lower, we showed that fusing 802.11 and GSM beacon readings can provide 100 percent coverage with a median error of just over 30 meters.

We also investigated how to bootstrap a beacon database from a partial database and keep it fresh as beacons are added or moved. We developed a graph-based self-mapping algorithm that lets a device with a small seed set of known beacon locations estimate the positions of unknown beacons while it locates itself.[2] Our three seed sources included the wigle.net public Wi-Fi database, commercial hotspot locators such as T-Mobile, and "opportunistic wardriving," in which GPS-equipped devices map out a few beacons any-time they can acquire a valid GPS lock. We tested self-mapping with nearly 100 days' worth of traces containing over 20 million beacon sightings and observed four results:

We also investigated how to bootstrap a beacon database from a partial database and keep it fresh as beacons are added or moved.

- Structured radio scans let us use as few as 10 percent of the beacons to bootstrap the beacon database.
- Accuracy depends on beacon distribution in the seed set. Uniformly distributed seed data outperform spatially clustered data four times as large.
- Both public beacon databases and sporadic GPS coverage are viable, real-world seeds, because neither requires users to manually map any beacons and both can ramp up to good coverage within a few days of normal use.
- With a seed of as little as 50 percent of the beacons, self-mapping can produce a radio map that estimates a user's location as well as a full wardriving database of all beacons.

We also studied place learning as a counterpoint to coordinate-based location such as latitude and longitude. We developed the BeaconPrint algorithm to enable a mobile device to learn the 802.11 and GSM response-rate histograms of places it's taken (home and work, for instance) and then recognize when it returns to those places.[3] We evaluated BeaconPrint using a month of trace logs from three people. When someone revisited a place, BeaconPrint correctly learned the place signature on the initial visit and accurately recognized subsequent revisits more than 90 percent of the time. When it erred, the percent of time BeaconPrint chose the wrong place was lower than previous approaches. The algorithm's greatest contribution is its success in learning and recognizing places visited infrequently or only for short durations. Accuracy is more than 63 percent for places the device has visited once and 80 percent for places visited twice.

We conducted three coordinated privacy experiments to understand users' feelings about disclosing their location to friends and family. The first experiment had a mobile device interrupt users periodically with simulated queries for their location.[4] In the second study, we built a mobile-phone-based location-aware system that let people exchange their location with friends and family; we piloted this on ourselves to compare with the previous simulation results.[5] Finally, we deployed the system to two recruited sets of families and their friends.[6] In each experiment, the results were broadly consistent: Users were keen to maintain direct control of their location information even though they rarely refused to disclose their location to people in their social network. Users were uncomfortable with both automated disclosure and automated refusal mechanisms. Finally, when we asked users to name their locations, they chose names beyond obvious descriptors such as "5th and Stewart St." They often chose names for clarity to the recipient. For example, a distant recipient would be sent "Seattle" rather than the street address. Similarly, some users included activities in place names—for

TABLE 1
Types of message threads in the Place Lab mailing list, number and percent of threads per category, and the percent of errors per platform prorated by the platform's download count.

Type of communication	Threads: Number	Threads: Percent of total	Threads: Percent of errors weighted by no. of platform downloads
Questions and comments about Place Lab (34%)			
Can I be independent of the Place Lab Web database?	8	21	–
How does algorithm X work?	8	21	–
Questions about accuracy	7	18	–
How do I map indoor beacons?	6	15	–
Requests for enhancements	5	13	–
Other	5	13	–
Platform-independent build or runtime errors (27%)			
I can't download/install/build	8	27	–
I can't get Place Lab to find its local data	7	23	–
The Place Lab Web database is down	6	20	–
I can't get my Bluetooth GPS to work	6	20	–
Other	3	10	–
Platform-dependent build or runtime errors (39%)			
Windows CE	23	52	54
Symbian Series 60 cell phones	8	18	27
Windows XP	8	18	6
Linux, OS X	5	11	13

example, "doing homework at home" or "on way to gym."

Finally, we scrutinized GSM beacon location in greater detail.[7] We found that GSM phones can achieve a positioning accuracy with a median positioning error of 94 to 196 meters using a single service provider's cells as beacons. Accuracy varied by a factor of four across different classes of algorithms, compared to an algorithmic variance of only 20 percent across algorithms with Wi-Fi. We also showed that wardriving only 30 percent of city streets for GSM provided positioning accuracy comparable to driving every street (60 hours of driving can map a metropolitan area such as greater Seattle). Finally, we showed that using three service providers' cells as beacons reduced the median error to 65 to 134 meters, which meets Enhanced 911 accuracy targets in the downtown area.

Lessons from mailing lists

Looking back on our research results and experiences (including having students lured away as consultants and having a car accident while wardriving), we learned a few lessons about deploying pervasive computing systems in the real world. These aren't lessons about location technology per se but practical things that would be out of place in a typical conference or journal paper. (See the "Related Research" sidebar for a list of other location research retrospectives, many of which focus on the technology itself.) We share these lessons here in the hope that they'll benefit other pervasive computing efforts, location-enhanced or otherwise.

One repository of collective knowledge about a system is the set of mailing lists and newsgroups that the system's users subscribe to. The placelab-users mailing list on SourceForge has 150 subscribers. In the past two years, 85 individuals (excluding project team members) posted more than 400 messages comprising 170 nonspam threads, with approximately half receiving a reply. Threads with more than one message had an average of slightly more than three messages, with a maximum of 11. The threads cover various topics; in this article, we focus on the set of 100 threads asking questions or reporting problems. To extract common experiences from these messages, we grouped them by topic and came up with the organization shown in table 1. This table breaks the users' communications into three roughly equal-sized categories: general questions about Place Lab, platform-independent problem reports, and platform-dependent problem reports.

Allow freedom from infrastructure

Place Lab lets devices estimate their positions without a network connection to any infrastructure. However, to encourage our lead users (who also wardrive and conduct location research) to share their wardriving data, we built a central beacon repository and a Web service to process wardriving logs. We believed our lead users would tolerate this small infrastructure dependency to gain the benefits of sharing their sets of discovered beacons. To encourage lead users to contribute to the central repository, we heavily promoted our Web-based beacon repository in our documentation, whereas we did not promote the command-line tools for converting wardriving data directly into beacon databases (although they were present in the Place Lab download).

Table 1 suggests that we might not have made the right choice. First, we didn't achieve 100 percent uptime for the Place Lab beacon repository. Even more important, the most frequently asked question about Place Lab was how to process wardriving logs and build beacon maps locally. So, we learned not to expect lead users to depend on the research infrastructure; they prefer to stay in control of their progress using a new system. From a practical viewpoint, you should create dependencies on research infrastructure only when necessary, and keep that infrastructure stable and operational at the level expected of a commercial infrastructure. This will allow the user community to flourish.

Bind code and data

The second-most common platform-independent error shown in table 1 is "I can't get Place Lab to find its local data." Place Lab, like most other pervasive systems, contains code to represent its behaviors and data to describe a deployment's particulars. Place Lab depends entirely on its data because it can't estimate location without a beacon database. To keep Place Lab flexible, we made it file-oriented: by specifying a path as a command-line argument, users can launch it with different data configurations. By default, if no path was explicitly stated, Place Lab would look for its data in a well-known location. While flexible, this mechanism unfortunately enabled the runtime error in which Place Lab's code lost track of its data. Our mailing list data suggests that this occurred often. In most cases, the error was something simple, such as the user placing the new beacon database in /My Documents/placelab/database rather than in /storage card/My Documents/placelab/database.

For real-world pervasive research, making your system work on all the latest devices requires a significant commitment of development.

So, you should have your pervasive application search automatically for its critical configuration data (or better yet, be bound to it such as in a segmented binary). Although less flexible and possibly less efficient, this approach is likely a good trade-off to help new users get up and running quickly with fewer errors.

Age before beauty

Platform-dependent errors outnumbered platform-independent errors (39 versus 27 percent). This isn't surprising: we wrote the platform-independent Java code only once, so it benefited from being tested and debugged across all the platforms. We had to rewrite the platform-dependent code to access the radios for each device.

The platform-dependent errors weren't distributed uniformly—over half the issues pertained to the Windows CE version of Place Lab. Counting the errors by message thread is misleading, however, because people didn't download the various versions of Place Lab in equal proportion. To account for this, we recalculated the fraction of errors reported for each platform, prorated by each platform's download count (see table 1, column 3). Assuming that the use of Place Lab across platforms is proportional to the download rate, we see a clear trend: Platforms had errors in roughly inverse proportion to their deployed base's size and the radio APIs' maturity.

For real-world pervasive research, making your system work on all the latest devices requires a significant commitment of development and support resources compared to making it portable across more mature platforms. For example, we probably could have saved resources by not trying to support a wide variety of Windows CE devices. Unfortunately, as our mailing lists show, many Place Lab adopters wanted to use the latest devices being released with Windows CE. This presents a challenge in balancing competing interests. Although core computing is standardized, pervasive computing inherently works on the edges where standards lag, and pressure to differentiate products causes devices to have substantial churn in their hardware and software interfaces.

Conventional wisdom

Of the many things we learned during the project, some in retrospect seem obvious—more like conventional wisdom than new lessons. While not new, three of these bear repeating as they might help you avoid common pitfalls and inefficiencies in deploying real-world pervasive systems.

Wake up and smell the flash

Devices running Place Lab need a local database of beacon positions. Without the database, no location estimation is possible. With urban GSM

Related Research on Location-Enhanced Computing

Because location-enhanced computing is a popular research topic, several other projects have published retrospective papers. Unlike this article, many of these focus on specific lessons about location technology or location-enhanced applications.

MIT's Cricket project presented a discussion of the basic hardware and software interface requirements for building accurate indoor location systems.[1]

AT&T's Sentient Computing group published an overview of the Active Bat approach to indoor ultrasonic positioning along with a more general discussion of the capabilities and possibilities of a highly accurate indoor location system.[2] Active Bat researchers also published a general set of lessons about deploying infrastructure and mobile devices for indoor environments.[3]

Another project was in the location-enhanced application space. On the basis of lessons learned in the Guide project, researchers from Lancaster University offered a set of guidelines for developing tourist applications for mobile devices.[4]

In robotics, a deployment of a mobile robot in the Deutsches Museum Bonn resulted in a paper on algorithms for location estimation, tracking, and robot motion planning as well as guidelines for managing human-robot interactions in a museum environment.[5]

Several papers evaluate and compare location algorithms for different sensing technologies and application scenarios.[6–9]

The ParcTab retrospective introduced the term "ubiquitous computing" and discussed the ParcTab hardware, networking, applications, and adoption by Xerox PARC researchers.[10] One lasting lesson from ParcTab is the importance of coverage. ParcTabs were useful devices, but their adoption into daily life was limited because they functioned only with infrastructure installed in the Xerox PARC Computer Science Lab and the homes of a few PARC researchers. This correlation between availability and adoption, which the ParcTab work suggested, was a primary motivation for us in Place Lab to enable a ubiquitous location capability.

REFERENCES

1. H. Balakrishnan et al., *Lessons from Developing and Deploying the Cricket Indoor Location System*, MIT Computer Science and AI Laboratory, 2003; http://cricket.csail.mit.edu/V1Exp.pdf.
2. M. Addlesee et al., "Implementing a Sentient Computing System," *Computer*, vol. 34, no. 8, 2001, pp. 50–56.
3. R.K. Harle and A. Hopper, "Deploying and Evaluating a Location-Aware System," *Proc. 3rd Int'l Conf. Mobile Systems, Applications, and Services* (Mobisys 05), ACM Press, 2005, pp. 219–232.
4. K. Cheverst et al., "Developing a Context-Aware Electronic Tourist Guide: Some Issues and Experiences," *Proc. Sigchi Conf. Human Factors in Computing Systems* (CHI 00), ACM Press, 2000, pp. 17–24.
5. W. Burgard et al., "Experiences with an Interactive Museum Tour-Guide Robot," *Artificial Intelligence*, vol. 114, nos. 1–2, 2000, pp. 3–55.
6. P. Bahl and V.N. Padmanabhan, "Radar: An In-building RF-Based User Location and Tracking System," *Proc. 19th Ann. Joint Conf. IEEE Computer and Comm. Societies* (Infocom 00), vol. 2, IEEE Press, 2000, pp. 775–784; http://ieeexplore.ieee.org/xpls/abs-all.jsp?arnumber=832252.
7. Y.C. Cheng et al., "Accuracy Characterization for Metropolitan-Scale Wi-Fi Localization," *Proc. 3rd Int'l Conf. Mobile Systems, Applications, and Services* (Mobisys 05), ACM Press, 2005, pp. 233–245.
8. D. Fox et al., "Bayesian Filtering for Location Estimation," *IEEE Pervasive Computing*, vol. 2, no. 3, 2003, pp. 24–33.
9. J. Krumm and E. Horvitz, "Locadio: Inferring Motion and Location from WiFi Signal Strengths," *Proc. Int'l Conf. Mobile and Ubiquitous Systems: Networking and Services* (MobiQuitous 04), IEEE CS Press, 2004, pp. 4–13; http://ieeexplore.ieee.org/xpls/absprintf.jsp?arnumber=1331705.
10. R. Want et al., *The ParcTab Ubiquitous Computing Experiment*, tech. report CSL-95-1, Xerox PARC, 1995.

tower densities of tens per square kilometer and urban 802.11 AP densities of hundreds per square kilometer, one of the most common questions we were asked was, "What research are you doing to help Place Lab devices manage the storage of their local beacon database?" People suggested various strategies ranging from caching (at the city, country, or continent level) to subsampling the database (to reduce its size without significantly impacting location accuracy). In reality, managing storage space isn't a significant research challenge in Place Lab. By our estimates, an indexed database containing all the cell towers in the world would require 125 Mbytes of storage, and adding all 802.11 access points grows it to 2.5 Gbytes. At the time of this writing, 1-Gbyte flash memory cards for phones are selling for US$40, meaning $5 worth of flash solves the Place Lab storage problem for GSM devices. The cost is even lower for a laptop with a disk. Ultimately, neither storage capacity nor processor speed is a bottleneck in Place Lab (although they're not completely irrelevant because they correlate with truly limited resources, such as battery power). We as pervasive computing researchers need to focus our resource management ingenuity on the right resources, such as battery power and users' attention spans, not just the traditional ones.

Keep it simple

Because the real world is complex, we assumed that vital parts of Place Lab would require complex solutions—for example, applying the latest machine-learning algorithms to model signal propagation. In mid-2003, we were building the first version of the Place Lab toolkit. We had a beacon database and could scan for beacons within range, but we needed an algorithm to combine the observations into a location estimate. We opted to start with the simplest algorithm we could think of: average the observed beacons' locations. Each beacon's location was a latitude-longitude pair, so the geometric effect was to place the user at the centroid of a polygon whose vertices were the beacons. This approach ignores many things that seemingly could improve the location calculation, including beacon signal strength, confidence in the beacon locations, or terrain features such as hills or buildings that block or reflect beacons' signals. Although the Centroid algorithm worked reasonably well, we considered it a placeholder that we'd replace shortly with a better one.

Throughout 2004 and into 2005, we invested considerable effort trying to improve the accuracy of Wi-Fi-based location using more environmental information and more sophisticated algorithms. We came to call these efforts "tinkering with the Tracker." We're chagrined that the hundreds of hours we spent tinkering never yielded a real-world improvement of more than about 20 percent despite being vastly more complex (and hence more error-prone) than our initial three-line algorithm.

In hindsight, we see the merit of simplicity. Using the Centroid algorithm's "80 percent solution" met Place Lab's non-accuracy-related research goals adequately. It was easy to comprehend in the toolkit, and its accuracy was sufficient to build applications, give technology demonstrations, and conduct multiple-user studies. This isn't to say that complexity is never necessary. For example, we concluded that Centroid is decidedly suboptimal on mobile phones with only GSM beacons, where a more complex Gaussian process algorithm is substantially better.[8] However, a simple, comprehensible solution providing most of the benefit might be good enough—going for the last 20 percent isn't always worth the effort.

Commodity hardware is easy to get

We designed Place Lab to work on unmodified commodity hardware. Although the mailing-list analyses we discussed earlier highlight the challenge of supporting less established platforms, supporting multiple commodity devices was a good choice for Place Lab because it let us do large deployments and build and sustain a user community. A software download from www.placelab.org and a trip to a consumer electronics store to buy a device are all you need to start working with Place Lab. Beyond the obvious ease of acquisition, using off-the-shelf devices for our research had two other effects.

A software download from www.placelab.org and a trip to a consumer electronics store to buy a device are all you need to start working with Place Lab.

First, research software running on unmodified commodity platforms doesn't draw huge crowds in a room full of ubiquitous computing demos. Because Place Lab uses everyday devices, there are no sleek new gadgets or custom circuitry to display. As a result, it lacks the "gee whiz" factor that people in the popular press, industry, and even academia sometimes look for to get turned on by a new idea. However, demonstrations of Place Lab running on real people's devices proved to be quite powerful. Seeing a location capability added to a device you already own makes it much easier to imagine how location technology might affect your life.

The second issue with using commodity platforms arose while doing user studies. To conduct our field work, we had two basic choices: Give the user a mobile phone just for running Place Lab or add Place Lab to their existing phone. (A variant of this second choice was to move their SIM and address book to a phone already running Place Lab). The first approach is clearly suboptimal, as most people own a mobile phone and don't want an extra device to lug around and keep charged. The second approach is much better from a study standpoint because it more realistically integrates our software with participants' communications and device-usage practices.

This approach has a subtle but important fault, however: it puts experimental software on the participant's critical communication device. Even though Place Lab is fairly robust, even one crash of the phone operating system a year represents a drastic decrease in reliability. In this case, the commodity nature of the hardware worked against us. We weren't giving the participant a strange, unique device. We were either giving them a second instance of something they only needed one of or an augmented but less reliable device they depended on. In the end, we opted for the reality of having the participants carry a single, location-enhanced device. We took extra care to show the participants how to recognize and restart a crashed phone, and one participant opted out of a study when she realized she might miss important phone calls in the event of a system crash.

Unconventional wisdom

Other, less conventional Place Lab lessons are probably more controversial but still useful.

The attention we gave to privacy concerns paid off in Place Lab, and we believe other researchers could similarly benefit.

You don't have to solve the ontology problem

An oft-cited, hard problem in location-enhanced computing is developing a common catalog of place names. A similar problem occurs in other ubiquitous computing domains in which there are ambiguous, overlapping taxonomies for activities, objects, goals, and so on. Investing the time to develop a good ontology might simplify automatic reasoning and provide compatibility across systems and versions. However, we offer two lessons from Place Lab:

- Solving the ontology problem isn't a prerequisite to doing good research studies on a topic such as place sharing (in fact, such studies should arguably come first).
- The lack of an ontology is preferable to an incomplete one, which risks user dissatisfaction or even abandonment due to a perceived lack of representational power.

Because no proven place name ontology was available, we chose to avoid hunting for a comprehensive ontology and instead compelled users to employ their own nomenclature. Our study participants used widely varying place names including "making sandwich at home," "work," "in transit," "bus stop," and "Canada." As our participants showed, people are remarkably sophisticated at using names to convey subtle and personalized meaning; indeed, this sophistication is a principal reason that the ontology problem is hard. Although we made users do extra work to invent names and we subjected ourselves and our users to sometimes ambiguous place names, users felt that the freedom allowed natural interactions. Without an ontology, we were still able to successfully test all the privacy and usability hypotheses in our three field studies.

You do need to talk about privacy

We addressed privacy directly in Place Lab, and in retrospect this was a good choice. We were up front about the project's privacy risks, benefits, and implications in all our publications, presentations, and deployments. In fact, the idea of privacy was so ingrained in our research that we jokingly considered subtitling Place Lab "The Privacy-Aware Location System for Privately Computing Your Private Location." Place Lab doesn't rely on external infrastructures that might surreptitiously record or reveal users' locations without their consent. This was a key architectural choice that we believe was responsible for Place Lab being one of the few location systems to avoid privacy-related criticism in any of the popular-media articles about it. We also studied privacy issues explicitly through a sequence of user studies of people's behavior with technology that let them easily share their location.

The attention we gave to privacy concerns paid off in Place Lab, and we believe other researchers could similarly benefit. A negative privacy reputation might be one of the quickest things to sour real-world users to a technology, particularly a new mobile or ubiquitous computing technology.

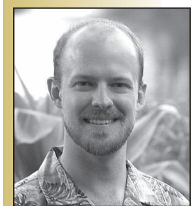
Context isn't just for applications anymore

As many user studies do, we interviewed participants before, during, and after deployment. In these interviews, users self-report how they did or didn't use the system and how the technology impacted their lives. One of the Place Lab user studies evaluated a mobile phone application that let users quickly and easily share location information with friends and family. Because the phones ran Place Lab, users didn't need to key in their location; the mobile device sensed it. Participants granted us permission to log the places they went and the content of messages they exchanged.

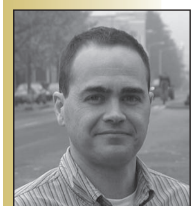
In one post-deployment interview, a participant described the numerous location notifications she had sent and the reasons why she shared her location. Her logs, however, showed very little use and substantiated almost none of the scenarios she described to us. (We verified that the logs were accurate and that there had been no hardware errors or data loss). Cross-validating with the real device context enabled us to understand the participant's willingness to contribute, and we could categorize her comments as being about how she saw herself using our system as opposed to how she actually used it.

The unconventional lesson here is that the real-world context captured on mobile devices can be as useful in conducting user studies and interviews

the AUTHORS



Jeffrey Hightower is a senior research scientist at Intel Research Seattle. His research interests include devices, services, sensors, and interfaces that help computing fade into the background of daily life as well as location-enhanced mobile computing. He received his PhD in computer science and engineering from the University of Washington. He's a member of the ACM and IEEE. Contact him at Intel Research Seattle, 1100 NE 45th St., 6th floor, Seattle, WA 98105; jeffrey.r.hightower@intel.com.



Anthony LaMarca is associate director of Intel Research Seattle. His research interests include location technologies, ubiquitous computing, distributed systems, and human-centered design. He most recently led the Place Lab project, which sought to enable wide-scale device positioning using radio beacons. He received his PhD in computer science from the University of Washington. Contact him at Intel Research Seattle, 1100 NE 45th St., Ste. 600, Seattle, WA 98105; anthony.lamarca@intel.com; www.intel-research.net/seattle.

Ian E. Smith is a senior research scientist at Intel Research Seattle. His work focuses on having a big bowl of ubicomp technology, social science, and some software engineering. Stir vigorously and don't forget to drizzle on some privacy. He received his PhD in computer science from the Georgia Institute of Technology. Contact him at Intel Research Seattle, 1100 NE 45th St., Ste. 600, Seattle, WA 98105; ian.e.smith@intel.com.

as it is in building applications. While traditional computing devices can log basic interactions such as messages sent and applications run, real-world pervasive devices can log much richer data during a user trial. Although this use of data clearly raises privacy issues, we believe that by being careful in the technology design, being up front with the participants, and putting them in control during the study, researchers can use the rich context data available in pervasive computing environments to help filter, correct, and understand users' self-reports.

The Place Lab approach of running on commodity devices, using existing infrastructure, and supporting existing applications resulted in a location technology that is practical for deployment in the real world. The core idea of the beacon-location approach was even commercialized twice during the life of our project, and Place Lab was occasionally criticized as being too practical for good research. We argue, however, that Place Lab is an example of how commercializable ideas can lead to high-impact research projects. Precisely because Place Lab's core idea is simple and straightforward, we had the opportunity to deploy a working system, experiment with different platforms, and investigate the research challenges that typically prevent systems from being practical in the real world. We were able to study privacy trade-offs, do field studies of new usage models, and add an elegant autonomy to our system by developing a self-bootstrapping beacon database. It's worth noting that neither of the commercial beacon location offerings supports features we see as the key Place Lab innovations, such as self-mapping, maintaining user control over private location data, or detecting the places the user visits.

Clearly, Place Lab's real-world nature was a net benefit to our research. We hope the lessons we learned can help future pervasive computing research seeking real-world deployment. P

REFERENCES

1. A. LaMarca et al., "Place Lab: Device Positioning Using Radio Beacons in the Wild," *Proc. 3rd Int'l Conf. Pervasive Computing* (Pervasive 05), LNCS 3468, Springer, 2005, pp. 116–133.
2. A. LaMarca et al., "Self-Mapping in 802.11 Location Systems," *UbiComp 2005: Ubiquitous Computing,* LNCS 3660, Springer, 2005, pp. 87–104.
3. J. Hightower et al., "Learning and Recognizing the Places We Go," *UbiComp 2005: Ubiquitous Computing,* LNCS 3660, Springer, 2005, pp. 159–176.
4. S. Consolvo et al., "Location Disclosure to Social Relations: Why, When, and What People Want to Share," *Proc. Int'l Conf. Human-Computer Interaction* (CHI 05), ACM Press, 2005, pp. 81–90.
5. I.E. Smith et al., "Social Disclosure of Place: From Location Technology to Communication Practices," *Proc. 3rd Int'l Conf. Pervasive Computing* (Pervasive 05), LNCS 3468, Springer, 2005, pp. 134–151.
6. G. Iachello et al., "Control, Deception, and Communication: Evaluating the Deployment of a Location-Enhanced Messaging Service," *UbiComp 2005: Ubiquitous Computing,* LNCS 3660, Springer, 2005, pp. 213–231.
7. M.Chen et al., "Practical Metropolitan-Scale Positioning for GSM Phones," to appear in *UbiComp 2006: Ubiquitous Computing,* Springer, 2006.
8. B. Ferris, "Gaussian Processes for Signal Strength-Based Location Estimation," to appear in *Proc. 2006 Robotics: Science and Systems Conf.,* MIT Press, 2006.