The Under-Appreciated Dimension of Time in Location-Based Systems

Karen P. Tang[1,2], Jason W. Wiese[2], Jason I. Hong[2], Daniel P. Siewiorek[2]

[1]Department of Informatics
University of California, Irvine
kptang@ics.uci.edu

[2]Human-Computer Interaction Institute
Carnegie Mellon University
{jwwiese,jasonh,dps}@cs.cmu.edu

ABSTRACT
Empirical evidence from past studies has shown that current location-based services (LBSs) are not individually compelling enough to drive LBS adoption. To boost the adoption rate, researchers should consider an under-explored research area: the time dimension of location information. We provide examples of such LBSs that share past and future locations. We also describe several challenges for designing these types of LBSs.

Categories and Subject Descriptors
H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

General Terms
Design, Human Factors.

Keywords
Location-based services, adoption, design space.

1. Where Location Services Stand Today
In the 20+ years since handheld GPS units were first introduced on the consumer market, location-based services (LBSs) have evolved to include more than just navigation and traffic services. In mobile application stores from providers like Apple, we have already witnessed a growing number of LBSs [13]. For example, from Jun 2009 to Feb 2010, the number of location-based iPhone applications more than doubled (to ~5,800 LBSs). We see similar trends from providers like Nokia and Google too.

However, aside from wayfinding applications, LBS adoption has largely been underwhelming. Moving forward, we believe that location-based services should reframe their outlook on how best to frame their services towards consumers. The current landscape of LBSs reveal a design space that is extremely one-sided. In particular, an overwhelming majority of these applications focus on a specific type of location sharing, namely sharing of *current* locations (i.e. “where are you now?”). While users generally agree that these services can be useful [8], they have not yet propelled LBSs to mainstream adoption.

In fact, the LBSs that share current locations are arguably useful for only a few scenarios (e.g., for coordination or okayness checking). However, empirical evidence from past studies suggests that these services do not *individually* provide enough value to drive the overall LBS adoption. In this paper, we present an overview of current LBSs, examine areas that have been overlooked in the LBS design space, and provide suggestions of how to move forward to address these new areas. Specifically, we advocate that LBS research should look at what we believe is a relatively unexplored area: the *time* element of location information (i.e., past and future locations). While there has been some research in this area, we think that there are still many outstanding challenges which we outline here as research questions. We also posit that, by addressing these challenges, we can diversify the LBS landscape to support other types of location sharing. As we will demonstrate, by providing consumers with a *diverse collection* of useful LBSs, it is hopeful that we can help booster the overall adoption of LBSs.

2. A Look at the Design Space for LBSs
To better understand the current landscape of location-based services (LBSs), we surveyed 97 commercial LBSs, mostly from North America and Europe. While this set of LBSs is by no means an exhaustive list, we feel that it provides a realistic representation of the different LBSs that are currently available.

It is often useful to analyze the application design space as a way to better understand the missed opportunities in a particular domain. There are several different ways we could analyze our list, and these LBSs can be analyzed along many dimensions. In this paper, we discuss these LBSs using a new categorization that we believe reveals additional research opportunities. First, we grouped each of the LBS according to the type of scenario they supported, which we based on the self-provided description on their website about their services and features. Overall, we found that current commercial LBSs tend to support nine different location scenarios: people finder services, place finder services, recommender services, tagging places or media, carpooling, trip planning and sharing, and personal informatics (see Table 1). We found that an overwhelming majority of these LBSs tend to fall into the “people finder” category (70.1%).

“People finder” location-based services are also popular within the research. Based on field deployments of these systems, a consistent finding is that these systems never attain high usage. For example, after an initial recruitment of 650 users (from 2 field trials), the ActiveCampus system stabilized to only 25 users after a month, a third of whom were affiliated with the research project [6]. After a 2-week trial, Locaccino users received, on average, only one location request per day [15]. Similar results were found after a 4-week trial with imbuddy [7].

Application Type	Application Scenarios	Location Type	Commercial Application Examples	%
People Finder	Where are my friends now? What new people are near me?	Where am I now?	Aka-aki, AT&T FamilyMap, Bliin, Blummi, Brightkite, Buddy Beacon, BuddyMob, BuddyWay, Centrl, ComeTogethr, Dodgeball, EagleTweet, EarthComber, findme, Flaik, Fireball, Firebot, Foyage, Friends on Fire, GawkerStalker, Geo-Me, GeoUpdater, Glympse, Google Latitude, Groovr, iFob, iPling, Ipoki, Limbo, Locatik, Locle, Loki, Loopt, Map My Tracks, Match2Blue, MeetMoi, Meet Now Live, Microsoft Vine, Mizoon, Mobilaris, MobiLuck, Mologogo, MyGeoDiary, MyGeolog, Myrimis, Nowhere, Plazes, Pocket Life, Shizzow, Skobbler, Skout, Sniff, Sparrow, SPOT, Spotjots, The Grid, TownQueens, Trackut, Twibble, Twinkle, Twittelator, Uandme, Unype, weNear, WHERE, Whereis Everyone, Whrrl, Zhiing	70.1%
Place/Event Recommender	What restaurant should I go to? What interesting events are near me? Which bar is everyone going to?	Where am I now? (some: Where might I go?)	Blummi, buzzd, Centrl, Citysense, FourSquare, Foyage, Geo-Me, Groovr, Lightpole, Limbo, Loopt, Match2Blue, Metosphere, Mizoon, MobiLuck, MobileCierge, Mologogo, Moximity, Pocket Life, Rummble, Skobbler, Socialight, Troovy, weNear, Whrrl	25.8%
Tagging Media	Where was this photo/video taken? What other photos have been taken here?	Where am I now? (some: Where have I been?)	Blummi, Brightkite, Ipoki, locr, Loopt, Microsoft Vine, MyGeoDiary, MyGeolog, Myrimis, Pocket Life, Twinkle, Twittelator, Unype, Zonetag	14.4%
Tagging Places	What did I think about restaurant? What did I think about this tourist site? What happened at this particular place?	Where am I now? (some: Where have I been?)	Findbyclick, Lightpole, Metosphere, Microsoft Vine, Myrimis, Outside.in, Socialight, TownQueens, Trapster, Wikinear	10.3%
Place Finder	Which gas stations are around here? Where is the nearest Starbucks?	Where am I now?	Brightkite, EarthComber, GeoSpot, Nanonavi, Quiro, Snikkr, WHERE	7.2%
Trip Broadcasting	What cities will I go to on my next trip?	Where might I go?	Dopplr, MAPme, TripIt	3.1%
People Reminder	Who's been to the places I've been to? Who will I bump into I go here?	Where have I been? Where will I go?	Blummi, whereyougonnabe	2.1%
Trip Sharing	Where did I go to on my last trip?	Where have I been?	Blummi, MAPme,	2.1%
Carpooling	Who can give me a ride to work?	Where am I going to?	Carticipate	1.0%
Personal Informatics	How many places did I visit on my trip?	Where have I been?	BuddyWay	1.0%

Table 1. A sample (N=97) of commercial location-based services. The majority are geared towards people finder scenarios (70.1%). Note that the total percentage exceeds 100% since some LBSs support multiple scenarios.

These findings reveal at least three aspects of the LBS design space. First, the design space for current LBSs is very crowded. Nearly three-quarters of these systems alone emphasize the people-finder scenario. Second, based on research deployments of people finders, very few people maintain active accounts and those that do, use the system infrequently. While this may be an issue of critical mass, it may also be due to what is perceived to be a lack of useful LBSs. Lastly, based on the available LBSs, it is clear that there is no clear winner or preferred LBS *yet*.

These observations can be initially discouraging. Critical mass is often seen as a necessary factor for successful LBS deployment [9] and many LBSs struggle with addressing this issue. While there are certainly other factors at play (e.g., location privacy), one possible reason for users' lackluster response is simply because current LBSs are somehow not compelling enough. This is not to say that there is no need for people-finding location services, or that such services are not useful. Instead, we believe that, *when considered individually*, people-finder LBSs are simply not useful or compelling *enough* to propel mainstream LBS adoption. In other words, while there are certainly compelling scenarios for when it is important to find a person (e.g., for coordination purposes), the habit of looking up someone's location is not practiced enough to sustain high usage for LBSs. Based on this insight, we propose that there is an underutilized design dimension that should be further explored to create a location platform that is more appealing for users.

2.1 Underutilized Design Dimension: Time

Using the 9 location scenarios previously described, we can further classify the LBSs according to the type of location information they use. For example, in a typical people-finding LBS scenario, users must disclose their *current* location. In contrast, for trip sharing scenarios, users usually post details about where they went on a trip after the fact. In this scenario, users are sharing their *past* locations. For trip planning scenarios, users typically share which locations they would like to visit. Here, users are sharing their *future* locations that they are likely to go to or locations they hope they might soon go to.

With this perspective, we can view location more generally in terms of time: users can share either their *present*, *past*, or *future* location information. From our list of LBSs, we see that an overwhelming majority is designed to support a specific kind of location sharing, i.e., sharing of *current* locations. In contrast, relatively few LBSs support sharing of *past* and *future* location information. In the following two sections, we present potential location-based services that fall under these two LBS categories. We present these examples as a demonstration of the vast opportunities that remain to be explored in this space, if we consider redesigning LBSs to share location information in a more fluid manner that includes both *past* and *future* locations.

2.1.1 Opportunities for Sharing Past Locations

Collecting historical information can be as simple as logging a phone's location every time it is sensed. Thus, obtaining location traces is relatively straightforward, given the plethora of LBSs that are already capable of real-time capturing of users' locations. This presents a valuable opportunity for researchers: this data already exists and will only become richer and more complete over time. Our charge now is to understand what compelling applications are now possible, given that users will soon ubiquitously have such datasets. We present four novel location-based system ideas that use past location information.

Supporting anonymous connections. One opportunity for "past LBSs" is to provide interaction opportunities for people that have previously overlapped in time and space, and would target users who would have otherwise missed out on connecting. An example of this scenario can be seen in the "Missed Connections" section of Craigslist [3], where people post messages about encounters they may have had with others, but were unable to follow up with. Metropolitan cities like New York and San Francisco often generate hundreds of missed connection posts in a single day [2]. These posts are mostly romantic in nature, but one can also find posts for lost items (e.g., "keys founds at the coffee shop here"). In addition, this type of LBS can also be useful for scenarios like: needing to find witnesses for incidents at a specific location, or looking up a conference attendee that you previously crossed paths with.

Icebreaking. Another opportunity for "past LBSs" is to help people find common grounding during their interactions, e.g., when someone introduces himself to a new acquaintance or when building a relationship with a new friend. In these situations, a useful conversational grounding could include which places people have in common, e.g., "we both go to the same coffee shop" or "we have both been to Scotland". An analogy for this social connection can be seen when people wear branded T-shirts. By wearing a t-shirt that refers to Scotland, a person projects information about himself to those around him, which others may then react to and socially engage with him. Thus, this LBS provides a virtual "T-shirt" of sorts and offers a way to send social signal to others, which has been shown to be helpful when building new relationships in a community [4].

Finding Expert Advice. A third opportunity is to better support users who seek expert advice about a particular place. For example, when planning a vacation, travelers may be unsure of which places to visit. A travel site like Fodor's [5] is an example of an online community that has multiple posts asking for feedback on upcoming travel itineraries. Not only is finding experts a difficult task, it is also difficult to know the context in which the feedback is provided for (e.g., whether the expert has similar travel interests to the poster, as evidenced by their past travel itineraries and activities). In this LBS, users can use past locations of others to find an appropriate expert that is both knowledgeable and with whom they share common interests.

Self-Awareness. Another opportunity for "past LBSs" is to support personal reflection of a person's location history. One relatively low-hanging fruit for such reflection is to leverage location history in a recommender system. For example, the system might suggest to a user that he go to Caribou Coffee instead of Starbucks, since he has already been to Starbucks several times in the past week. However, this is a relatively limited use of location history. Instead, we suggest LBSs focus more on personal informatics, a field that focuses on providing users with information about themselves so that they can reflect, learn, and possibly change their behaviors. In the case of location, users might use their history to reflect on issues related to time management or work-life balance. For example, after examining his location history, a user could conclude that the hours spent working at a local coffee shop is not the most productive location for him. Or, on a weekly basis, a user may see that they spend too many hours at work and decide to try to reduce their hours to something they feel is more reasonable.

2.1.2 Opportunities for Sharing Future Locations

"Future locations" can refer to explicit plans, implicit routines, or plans to visit a place. While this data can be difficult to collect, many new opportunities are afforded if LBSs could support sharing of future locations. We present two examples.

Providing Smarter Notifications. Assuming that a user's location routines can be learned over time (e.g., [11]), "future LBSs" could be designed to provide better notifications for users. For example, every Tuesday, a parent may regularly travel from home to soccer practice to pick up their child. If the system can learn this routine, the LBS can preemptively notify parents that they may need to leave early to allow time for an impending traffic jam and to avoid late pick ups. Past work has shown that transportation coordination is an important area of burden and responsibility, especially for dual-income parents [1].

Planning for Overlap. Another opportunity for "future LBSs" is to better facilitate discovery of future interaction opportunities with others. For example, an LBS could allow for more dynamic ridesharing. Instead of requiring users to explicitly define routes that they normally take (e.g., from home to work and vice versa), the system could facilitate ride-sharing during other trips as well (e.g., when you're driving to the grocery store, to the shopping mall, etc.). Another possibility for this LBS is to encourage users to plan for more social experiences than they would have otherwise engaged in. For example, if a student realizes that his classmates will be studying in the library tomorrow night, then he may rearrange his schedule so that he can have a more collaborative study session.

In summary, we believe that carefully analyzing LBSs based on time (past, present, future) can reveal interesting and novel opportunities that the LBS community has thus far overlooked. This is not to say that current LBSs are not useful, but that they have not *individually* been useful enough yet for widespread LBS adoption. By considering other types of LBSs and the element of time, we can expand the design space to include new and potentially useful LBSs that share past and future locations.

3. Rethinking LBS Adoption

The design space for LBSs is much larger when we consider the fluidity of location information across time. Just as there is no single "killer app" among current LBSs, we also believe that no single "past LBS" or "future LBS" will be a killer app. Other than navigation services, many LBS-supported scenarios (current and proposed) simply do not occur often enough or are individually not critical enough to garner critical mass by itself. However, an important observation is that, together, these less-critical scenarios can still build critical mass. In other words, the key is that these scenarios must be packaged together for the user, so that, *as a whole,* LBSs can entice users to share their location information more often than any individual application would. Thus, the focus should not be on implementing a single killer app, but rather on building a framework to support a collection of LBSs. This approach leverages the fact that, while a single LBS may not be used often, the *collective usage* of a suite of applications (spanning past, present, and future locations) can encourage widespread LBS adoption by helping to increase the overall perceived value of LBSs for users.

We propose that the LBS community should rethink the framework in which their services are deployed. From Table 1, we can see that only a few LBSs support more than a handful of use cases. In an effort to facilitate easier development of LBSs, frameworks like Yahoo's Fire Eagle [17] are becoming increasingly popular among developers. Fire Eagle maintains location information for a particular person and mediates access to it by other applications. It allows users to select applications that they would like to pull location information from (or push location information to) and provides developers with an easy to use location-related application programming interface (API).

While this framework is a step in the right direction, it lacks certain features that make it difficult to implement a unified approach for providing a true platform of diverse LBSs. First, Fire Eagle is a centralized system. An application on the user's phone reports their location to the Fire Eagle server or polls the server for the most recent location information. With this reporting model, the fidelity of Fire Eagle's location information is controlled by whatever the most recently used LBS reported. Thus, when running concurrent LBSs, a potential problem is that different applications may need different levels of location granularity. This issue is further exacerbated when considering LBSs that use past and future location information.

Second, Fire Eagle has no resources for maintaining information about users' past and/or future locations, continuously updating the user's account with their most recent location. However, to support LBSs with different time scales, there needs to be: 1) some sort of repository for maintaining a user's past locations, and 2) some sort of logic for inferring a user's future locations.

Thus, we can see that Fire Eagle is currently ill-equipped to support the types of LBSs that we are proposing. However, without a framework, successfully building a streamlined user experience for a suite of LBSs will be difficult, not only from an application development perspective, but also from a usability perspective. In the next section, we describe various technical challenges involved in realizing such a framework.

4. Technical Challenges

Although there are many factors at play, one way to facilitate widespread adoption of LBSs is to expand the LBS design space. We specifically call out one particular design dimension that we feel is underutilized: time. But to properly support the development of these relatively new types of LBSs (i.e., ones that leverage *past*, *present*, and *future* locations), a new type of LBS framework is needed. Fire Eagle has led to a major increase in the number of LBSs sharing *current* location data. However, to support other location types, a new LBS framework is needed and several technical challenges need to be addressed.

4.1 Data

When designing a framework to support sharing past and future locations, several issues relating to data become important.

4.1.1 Data Storage

In Fire Eagle, most LBSs only need a data structure that contains the most recent latitude-longitude coordinates (usually two `doubles`) and a timestamp (usually a `long`). For "past LBSs" and "future LBSs", slightly different data structures are needed.

For example, to build LBSs that support sharing of past location trails requires, at a minimum, at least three data types: latitude-longitude coordinates, a timestamp, and a duration indicating how long the user has been at that location. In certain "past LBSs", it may be beneficial to also have a data type that stores the surrounding Bluetooth IDs (or some other unique identifier) to indicate who was around the user at that time and place. "Future LBSs" (like the smart notifications described earlier) would also require at least three data types: latitude-longitude coordinates, a timestamp, and an array to hold the expected route information for arriving at the specified coordinates.

These modified data structures introduce additional memory constraints. For "past LBSs", having the duration (usually a `double`) results in a 40% increase in storage requirements and can quickly add up when considering that streams of locations are recorded, not just the most recent location. This suggests that it is important to consider smarter location sensing algorithms to minimize over-sampling and redundant data so as to not waste data storage. In addition, effective access and retrieval methods are needed. For the anonymous connections example, there are many ways to compute commonalities. Not only is there overlap in the same location, there could also be overlaps in how long people stay at a place (e.g., you both grab take-out vs. dining in) or when people arrive (e.g., you both pick up afternoon coffees).

4.1.2 Location Abstractions

Many "current LBSs" represent location information as latitude-longitude coordinates. Some also let users specify a label to make the location information more readable to others. For "past LBSs" and "future LBSs", we need to consider which location abstractions are provide the most value for users. For example, in the anonymous connections example, do users prefer knowing that they: 1) both visit Starbucks, 2) both visit the downtown Starbucks, or 3) both visit the Starbucks at 100 Main St? Determining what is generally useful and what user are comfortable with sharing will directly impact the types of location abstractions that the system needs to understand and represent. Accurately and unobtrusively collecting these abstractions in a privacy-sensitive way is a challenge for LBSs.

For "past LBSs" like the self-awareness example, there is also the additional challenge that location abstractions may change over time. For example, users may find that they require more detailed location information for the past week (perhaps precise latitude-longitude coordinates), but, for data from a year ago, they may just need coarse-grained location information (e.g., semantic labels like business names or "home"). Thus, flexible support for a variety of location representations is important.

4.1.3 Location Visualizations

LBSs that share only current locations tend to visually present that information on a map. While this may be an intuitive way to represent latitude-longitude coordinates, the additional complexity of past and future location information introduces new opportunities for other types of information visualizations. However, there are several important design issues when considering these visualizations.

First, there is the issue of privacy. Different visualizations may lead users to feel more or less comfortable with the type of information they are disclosing. This is particularly relevant for "past LBSs" where a user may be sharing large amounts of data (many locations vs. one location) in a single transaction. Second, these visualizations must consider the utility of the visualization as well. While the map may be the easiest way to represent the information, it may not the most *useful* presentation to others. For example, in "past LBSs" like the self awareness example, it can be difficult for readers to extract from a map certain trends, like how often you go to a place or which days you visit a place.

Designing and evaluating alternative information visualization designs can help draw out these trends more clearly.

4.1.4 Data Consistency

It is unlikely that users will have location information available 24/7. For example, phones are physically turned off on airplanes and may be off while sleeping at night. During these times, there will be gaps in the locations being recorded. In addition, inaccuracies in sensing data can lead to erratic location reports, particularly when using non-GPS location sensing or when using GPS sensing while indoors. These challenges are, in fact, applicable for "current LBSs". However, they become more problematic when considering "past LBSs" and "future LBSs".

The magnitude of these concerns depends on user expectations. For example, in "past LBSs" like the self-awareness example, it is unclear how detailed of a location history they require. Will gaps in the data be problematic? In the data gap generated by a flight, would users prefer to have location trails representing their in-flight movement? Obtaining this information is certainly feasibly (e.g., through querying flight tracking services), but it also presents additional complexity for designing these LBSs. For "future LBSs", having gaps in the data can make inferring routines much more difficult or may lead to incorrect inferences.

It may be possible to overcome some of these data gaps. For example, when gaps are from poor GPS coverage (e.g., entering a building or an area with several surrounding buildings), the device can switch to an alternative location-sensing algorithm. When gaps are from having no Internet access, devices can temporarily locally cache location information until a data connection is reacquired. Even when Internet access is available, users may not want to upload all of their data. This could be for power consumption reasons (making many GPRS connections can drain the battery) or because of a slow connection (3G connections are not always available and EDGE connections often consume less battery power than 3G connections). In these cases, offline caching can also be useful. Thus, a challenge for LBSs is to develop algorithms for efficiently switching between different sensing algorithms, as well as for intelligently switching between offline and online data connections.

4.2 Centralized Approach

In a purely centralized LBS approach, we assume that there is a very thin client that mainly provides a front-end UI. All sensing is done using existing infrastructure (i.e., not on the device) and most of the processing is done "in the clouds". In a mostly centralized approach, the client also supports location sensing, but all location information is uploaded to third-party servers.

4.2.1 Privacy: Data Retention & Inferencing

One important challenge for implementing LBSs based on the centralized approach is how to maintain users' data privacy. In "past LBSs" like the anonymous-connection example, a location history is needed to determine whether there are overlaps between users. This means that the LBS must have access to location histories for many users, resulting in two significant privacy concerns. First, entire location histories are stored in one central repository. While information sources like a person's social security number has obvious privacy implications, the types of inferences that a person's location history may reveal is not always immediately obvious to the user. Prior work has shown that it is possible to infer a user's home and work [10] and the routes they normally take [12]. As more location information is gathered about a user, more of these inferences can be made. Thus, it is important that not only is this data securely stored, but also that we: 1) understand the extent to which users are comfortable in sharing this information with third-parties, and 2) minimize, or even prevent, unnecessary inferences to better protect the user's privacy. The challenge will be finding a solution that protects the user while still providing them maximum benefit from the LBS.

Another important privacy issue is data retention. How long should servers need to retain users' location history to support different scenarios? How far in the future should LBSs infer patterns? Currently, Google Latitude's retention policy states that locations won't be retained for more than 24 hours. Using this policy may adversely affect user experiences for both "past LBSs" (having less data reduces the utility of these applications) and "future LBSs" (inferencing is harder with less historical data). Thus, appropriate retention policies are needed to balance users' privacy concerns with LBS feature requirements.

4.2.2 Communication Overhead

In a mostly centralized architecture, additional network traffic is generated when the client device uploads its location reports to the server and when it downloads the information from the server. In a purely centralized approach, communication overhead is only needed for the downlink since location sensing takes place in the infrastructure, not the client device. By working to minimize this overhead, LBSs can further optimize the battery life of client devices. Not all LBSs require continuously updates of their location information. However, only updating location information when the users activates the UI may not be optimal, particular for "past" LBSs where the client may need to suddenly download large amounts of data. Thus, LBS designs should take these issues into consideration.

In summary, using a centralized approach for building LBSs enables us to offload much of the processing from the client. However, the challenge for this type of design is ensuring that users' data privacy is maintained, adequate security protocols are in place for storing the data, proper data retention policies are in place, and communication overhead is taken into account.

4.3 Decentralized Approach

In a decentralized LBS architecture, we assume a "fat" client that performs its own sensing, processing, and storage of location information. The resulting information that is uploaded to the server is thus the final location representation, which, depending on the application requirements, may be as precise as latitude-longitude coordinates or as abstract as a text label. The immediate advantage of a decentralized approach is that users maintain control of their own information, which significantly reduces privacy concerns for the user. However, the privacy benefits come at the cost of power and additional processing. It is also important to note that the client device is not always secure. Users can lose their device and an informed attacker can infer users' past locations, such as their home and work. Phones are also frequently replaced by newer devices. Just as recycling old hard drives introduces potential security vulnerabilities, reusing a phone that still contains the original owner's location history may expose sensitive information about the user.

4.3.1 Power Consumption

In a decentralized architecture, the already limited battery life of the mobile device is further taxed. This is exacerbated by the fact that collecting location history may require a certain frequency or granularity of information to be most effective, causing an even greater power demand. One approach, hierarchical power management [16], uses low-powered sensors to determine when it will be most effective to: 1) use high-powered sensors to perform more accurate sensing, and 2) perform more taxing processing tasks. While this can be helpful, further research is needed to extend these types of algorithms to consider application requirements for "past LBSs" and "future LBSs". For example, in "past LBSs", one could use prior locations to inform which location sensing method to use, which can potentially lead to significant additional power savings.

4.3.2 Computation Overhead

Increased computational loads for mobile devices can result in lower battery life. In a decentralized architecture, the client device is responsible for interpreting all of its sensed location reports. To properly support "past LBSs" and "future LBSs", efficient algorithms are needed to process potentially large data sets. For example, in "past LBSs", different inferences are possible when considering one day vs. one month of location data. Similarly, "future LBSs" can benefit from knowing about both similarities (e.g., for inferring information about routines) and differences (e.g., for determining coping mechanisms for routine breakdowns). Finding algorithms to quickly determine these features can drastically improve user experiences with these LBSs as the accuracy and processing speed of these abstractions often directly influence whether users end up adopting LBSs.

In summary, using a decentralized approach for building LBSs provides additional privacy benefits for users. However, the major challenges for this type of system design are optimizing battery life and processing overhead for the client device.

4.4 A Hybrid Approach

It is worth pointing out that choosing between a centralized and decentralized architecture can directly impact the types of LBSs one can deploy. For example, by keeping location information stored on the device (as in a decentralized approach), we significantly increase the complexity of certain LBSs. Consider a LBS that uses proximity information between two different users. In a centralized approach, each user reports their location information to the server, which can accurately determine the distance between different users and notify them when they are near each other. In a decentralized approach, since there is no central repository, the client device must now calculate proximity based on some other type sensing. One possibility is to use the Bluetooth signal strengths of other devices. However, this assumes that nearby devices have Bluetooth activated (which is seldom the case), and it requires the developer to have access to low-level protocols to compute the RSSI for Bluetooth.

Thus, for this example, the centralized approach clearly offers a much simpler LBS application design. The downside of a centralized approach is that it introduces significant privacy concerns for the user. However, it is not necessary for systems to be completely centralized or decentralized. We propose a *hybrid approach* that combines the benefits from both types of architectures. One possible hybrid design is to offload some data storage and processing to a centralized server. The client device would hold all the accumulated locations. The server would only contain short-term locations (e.g., less than a day's worth). The processing of the immediate locations can be done remotely, saving the client device battery life and computational overhead. In terms of privacy, the user still maintains more control over their location information than in a purely centralized approach, but there is still communication overhead needed to ensure that any data residing on the server is downloaded to the client before it is replaced with more recent location information.

A challenge in creating an appropriate hybrid design is evident when creating LBSs that aggregate location information. Consider an LBS that reports which places are popular, based on how many people visits that location. This requires some type of central repository. However, to preserve privacy, it will be important to incorporate anonymization techniques so that the shared information does not reveal a user's location habits. While past work has started to examine this (e.g.,[14]), further work is needed to extend this idea to other aggregate LBSs that will use and share past and future locations.

In summary, we describe four technical challenges that apply to building a new LBS framework to better utilize the time dimension of location sharing. We present these to the LBS community as a "call to arms" for researchers to consider alternative LBS designs. We also present a taxonomy of LBSs and several novel LBS concepts that demonstrate the potential utility for including past and future location support in next-generation context-aware information sharing systems.

5. References

[1] Beech, S., Geelhoed, E., Murphy, R., Parker, J., Sellen, A. and Shaw, K. 2003. Lifestyles of working parents: Implications and opportunities for new technologies. HPL-2003-88R1, HP.

[2] Cox, L.P., Dalton, A. Marupadi, V. 2006. Presence-Exchanges: Toward Sustainable Presence-Sharing. In HotMobile '06, 55-60.

[3] Craigslist. Missed Connections. http://craigslist.org/mis/.

[4] Donath, J. Identity and Deception in the Virtual Community. Routledge, City, 1998.

[5] Fodor's Forums. http://www.fodors.com/community.

[6] Griswold, W.G., Shanahan, P., Brown, S.W., Boyer, R., Ratto, M., Shapiro, R.B. and Truong, T.M. 2004. ActiveCampus - Experiments in Community-Oriented Ubiquitous Computing. IEEE Computer, 37, 10, 73-81.

[7] Hsieh, G., Tang, K.P., Low, W.Y. and Hong, J.I. 2007. Field Deployment of IMBuddy: A Study of Privacy Control and Feedback Mechanisms for Contextual IM. In Ubicomp '07, 91-108.

[8] Junglas, I.A. and Watson, R.T. 2008. Location-based services. ACM Communications, 51, 3, 65-69.

[9] Kaasinen, E. 2003. User needs for location-aware mobile services. Pers Ubiq Comp, 7, 1, 70-79.

[10] Krumm, J. 2007. Inference Attacks on Location Tracks. In Pervasive '07, 127-143.

[11] Liao, L., Fox, D. and Kautz, H. 2004. Learning and inferring transportation routines. In AAAI '04, 348-353.

[12] Patel, K., Chen, M.Y., Smith, I. and Landay, J.A. 2006. Personalizing routes. In UIST '06, 187-190.

[13] Skyhook Wireless. http://skyhookwireless.com/locationapps/.

[14] Tang, K.P., Keyani, P., Fogarty, J. and Hong, J.I. 2006. Putting people in their place: an anonymous and privacy-sensitive approach to collecting sensed data in location-based applications. In CHI '06, 93-102.

[15] Tsai, J., Kelley, P., Drielsma, P., Cranor, L.F., Hong, J.I. and Sadeh, N. 2009. Who's viewed you?: the impact of feedback in a mobile location-sharing application. In CHI '09, 2003-2012.

[16] Wang, Y., Lin, J., Annavaram, M., Jacobson, Q.A., Hong, J.I., Krishnamachari, B. and Sadeh, N. 2009. A framework of energy efficient mobile sensing for automatic user state recognition. In MobiSys '09, 179-192.

[17] Yahoo. Fire Eagle. http://fireeagle.yahoo.net/.