Simulating and Implementing Geospatially-based Binding Mechanisms for Mobile Peering

Ben Falchuk
Telcordia Technologies, Inc.
1 Telcordia Dr.,
Piscataway, NJ 08854 USA
bfalchuk@research.telcordia.com

David Shallcross
Telcordia Technologies, Inc.
1 Telcordia Dr.,
Piscataway, NJ 08854 USA
davids@research.telcordia.com

ABSTRACT
In mobile peer-to-peer information sharing, binding to - and sharing with - another device uses up computing resources and makes sense only when certain conditions are met. We propose a visual notation and an software tool for mobile devices that address the need for a simple and intuitive way to allow the setting and testing of policies for on-the-go information exchange (e.g., playlists). This intuitive and visual approach - called MotionMaps - is well-suited for small screens and helps ensure efficient use of device and network resources. This last claim is supported in this paper by simulation and analysis that makes clear the impact of these policies on the effectiveness of information exchange across several mobile use cases.

Categories and Subject Descriptors
H.5.2 [**User Interfaces**]: *screen design, user-centered design*, C.2.1 [**Network Architecture and Design**]: *wireless comm.*.

Keywords
Peer to peer communications, On-the-go information sharing.

1. INTRODUCTION
The mobile distributed computing paradigm, in which communications devices are on-the-go and enter into ad hoc peer-to-peer communications with other such devices, is more and more prevalent. Recent and newly widespread underlying consumer electronics technologies making this possible include: Global Positioning System, IEEE 802.11 (Wi-Fi) and Bluetooth (now shipping with a majority of new smartphones), rich graphics and user interfaces capabilities (e.g., Java2ME), availability of dual-mode (Wi-Fi/cellular) smartphones, and middleware such as that from the OSGi Alliance that allows a device to use software components as they become available, without a restart or disruption. Thus, ad hoc device discovery and binding are real, as are rich services and applications. Currently, a large number of mass-market mobile devices are overtly intended for consumers who want to not only carry their media with them at all times but also connect opportunistically to local networks and peers. Most notably, this device-class includes Microsoft's Zune™, whose WiFi capabilities allow and *encourage* playlist and song sharing amongst nearby users. Other devices include ultra-mobile PC's by Apple, Wibrain, and Kohjinsha. In addition to mass-market hardware trends, social networking trends which have pushed the Web into the so-called Web2.0 phase are now making the notion of sharing user-generated content between small mobile devices very desirable. True, some users remain concerned about privacy and security during these exchanges but studies have shown that passionate content-producers are more willing than ever to share content, demographics, lifestyle and personal tastes information despite the small but real risks of breach, malware, and so on [1]. Heavily trafficked Web sites reinforcing this notion include Flickr, Blogspot, YouTube, and MySpace, while BitTorrent, eDonkey and other file-sharing tools still comprise a great deal of all Internet traffic.

Our work addresses several key concerns and open issues in peer-to-peer connectivity for information sharing. First, even though wireless devices proliferate, none have simple, intuitive, user-centric tools that allow the fine-grain tuning of geospatially-based sharing policies. None offer users the ability to visualize and test these policies for6 mobile use cases. Secondly, in a mobile setting every act of sharing uses (and drains, in the case of battery) resources such as network capacity, computing capacity, and battery life. Thus, the current art of geospatially-*insensitive* exchanges actually sets up the device to *waste* valuable resources. Our work, then, has the following contribution and highlights:

- applies principles of interface design for small devices to the new and highly germane problem of geospatially-sensitive information sharing
- a visual "language" with which policies are built and tested
- proof-of-concept developed on Windows Mobile 5 device
- a simulation and analysis that convincingly draws out the benefits of using geospatially-sensitive binding policies to police peer-to-peer information sharing

2. GEOSPATIAL BINDING
Geospatially-based binding can be thought of as any binding methodology that includes - and takes into account - information about one or more of the following attributes of the "binder", the "bindee", or both: their position, their speed, or their direction, in either a relative or absolute sense. Human interactions intuitively follow geospatial rules: When bicyclists cross paths they usually do not begin what will be long conversations; instead they *anticipate* future resource availability and modify their exchange accordingly. With device-based information-sharing, both binding and information sharing consume valuable computing resources and this makes intelligent trade-offs, analogous to the human-case, all the more important. Consider the following: A device and its user, *C*, are on a bicycle. *C* is interested in sharing

AMBI-SYS 2008, February 11-13, Quebec, Canada

DOI 10.4108/ICST.AMBISYS2008.2866

her playlists and also in receiving playlists from people she passes near (as are *D* and *E*). As *C*'s device comes into range of *D* and *E* P2P sharing becomes possible between the three. However, geospatial properties of the devices - namely their relative speeds and relative directions to each other - should also come into play. Why? Because if *C* and a given peer are not in range of each other for long enough, no successful sharing session can occur; in this case *no sharing attempt should even be initiated* as it will abort and fail (as an incomplete transaction) the moment that the peer goes out of range. Additionally, binding should not occur if *C* and the peer do not have compatible sharing preferences or privacy settings. Hence four main dimensions can be instrumented to help ensure efficient resource utilization: 1) relative speed, 2) relative direction, 3) binding preference, and 4) shared information type. A tacit goal, then, is to maximize the number of successful sharing transactions made in a given time while avoiding those peerings that will *probably* result in interruptions or won't result in completed transactions. Consider the case of trying to share information in a busy street full of candidate peers. Without any geospatial tempering, the device is likely to use a greedy algorithm (first come first bound to); assuming a variety of speeds and wireless signal strengths the greedy algorithm is likely to result in the device continually binding to peers, beginning information transfer, and aborting when the peer goes out of range before the transfer is complete. Therefore, "return on information" with respect to battery usage is very low and a real practical application of our work is its embodiment into device middleware as a battery-usage optimizer.

Even though today's mass market devices are notorious for short battery-life, none presently support power-based binding for playlist, song, and favorites sharing. But this work has applicability in other realms too, including, but not limited to:

- Military - in dismounted troop movements it can be beneficial to exchange information only with those traveling in certain directions (e.g., are they on my trajectory or not?).
- Manufacturing - automated manufacturing floors employ both mobile and stationary robots for which directional-sensitive sharing may be useful.

Meanwhile, applications of positioning and trajectory are seen in: guidance systems (tourism), public safety, and aids for visually impaired users. This paper focuses on the following mass-market cases:

1. Walking on 5th Ave. – A user and device walk on a busy sidewalk. The user is mobile in a fixed direction and peers likely to be either in opposite or co-directional. Peers' speeds are fairly predictable.
2. Sitting on Campus Green – stationary on an active lawn. Peers approach from relatively random directions with a wide variety of speeds (e.g. on bikes, foot, etc.)
3. Walking in Grand Central Station – the user is mobile (on foot) and encountered peers from random directions with predictable speeds.

We study the merits of our approaches in each of these representative use cases[1]. Note that if we assume that media transfer times are small (more on this later) our technologies will tend to be less useful when very few peers are encountered per time interval, or when all parties are nearly stationary. In these cases, even without our technology the device will see good results by simply using a greedy approach.

[1] New mobility models [8] themselves are not the goal of this work

2.1 Metadata Exchange

We assume that a given device has access to the metadata it needs in order to decide go/no-go for a given exchange. Link level and service discovery protocols such as Universal Plug and Play, Service Location Protocol (SLP), Jini, and Salutation, employ registries and string-matching to match client's needs with registered services. Much past work has expanded registry/search semantics to allow for rich service querying - e.g., in [2] the authors extend Bluetooth discovery with additional semantics. In [3] the authors similarly extend the OSGi framework. 802.11 access points broadcast beacon frames about every 100ms; they are 50 bytes long and carry SSID, rates, timestamps and other information. In this paper we assume that devices have both a low-level way of emitting and detecting beacons (for inter-discovery) but also - since it will be a basis for decision-making - an efficient way of exchanging geospatial and other service metadata which may be comprised of:

- GPS (or other location) coordinates
- Instantaneous or average speed, direction
- Other service attributes - such as sizes of media that are ready to be exchanged

Our past work [4] has shown that a rich ontology (and parser) can be an enabler, at the cost of computational complexity. In such a scheme peer metadata is read as instances of ontology elements and interpreted accordingly. One advantage of this approach is that unit equivalence may be inferred (e.g., direction may be provided in any number of ways: compass direction (e.g., "NE"), a set of LAT/LONG's comprising a vector, degrees (e.g., "270 degrees"), etc. In this paper we are concerned only that a peer's geospatial metadata is understood, *not with the actual format or mechanism of this exchange.*

Table 1. Media and inter-device transfer times (TT)

Type	Size	TT for rate 0.5 Mbit/sec	TT for rate 10 Mbit/sec
Movie	500 MB	8000 sec (2.2 hr)	400 sec
Episode	250 MB	4000 sec (1.1 hr)	200 sec
Music video	30 MB	480 sec (8 min)	24 sec
Song	3 MB	48 sec	2.4 sec
Component	500 KB	8 sec	0.4 sec
Favorites	200 KB	3 sec	0.15 sec
Playlist	20 KB	0.3 sec	0.015 sec

2.2 Devices and media

Today's mobile smartphones, mobile media players, and ultra-mobile PC's (UMPC) have widely varying capabilities as their manufacturers strive to create design and feature combinations that appeal to market niches. A core set of functionality is emerging that ultimately embodies enough functionality to allow peer-to-peer sharing of various sorts. A key one is clearly WiFi networking, which enables peer-to-peer sharing at high speeds (Bluetooth is ubiquitous but has limited utility as a file-sharing transport and can experience interference near WLAN networks [5]). A mobile device with WiFi technology has varying effective ranges; for example a laptop with WiFi can expect about 90 feet of range while some users have

seen the Zune's range to be about 30 feet. Practical WiFi transfer rates also vary in real-world conditions. Other experimentation with Zunes has shown effective transfer rates to be about 0.5 Mbps [6]. Transfer rates are less important when the basic currency of exchange is a playlist or Internet favorites but they become a concern when the currency is songs and episodic television shows, for example. Today's consumers are already buying and sharing many types of information. Table 1 summarizes these types, their typical sizes, and estimated transfer times (TT) for several relevant bitrates.

Episode refers to episodic television such as NBC's 'The Office', or short films. *Components* refer to exchangeable, reusable software objects - within OSGi and other Java (and even non-Java) frameworks, software objects can be bundled and re-used across platforms in a very dynamic fashion [7]; e.g., a running application on one device may require an object present on another device (a specific media-player component, or temperature-converter) - OSGi provides the foundations for seamless run-time use of this object. *Favorites* refers to Internet browser favorites.

3. SIMULATION AND RESULTS

This section describes our simulation of the 3 use-cases and the merits of using geospatial policies to filter out mobile binding candidates that are "undesirable" while binding to those that are likely to result in successful media transfers. This section relies upon a visual metaphor with which filters will be presented.

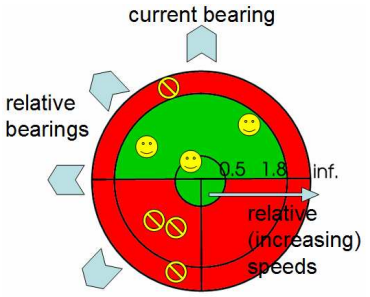


Figure 1. Simulation details are presented in a simplified MotionMaps style; a colored coordinate space

Filters are represented on a radial coordinate space in which the up-facing angle is a given users current bearing and the other angles correspond to relative bearings. Similarly, distance from the center indicates increasing relative speed. Figure 2 shows 7 peers mapped onto a user's filter-space - three (smiley faces) map onto green regions and the filter lets these pass through to the binding phase, while four map to red regions indicating their undesirability for binding (either because their speed or angles are large).

We implemented a discrete-event simulation to evaluate the effect of using filters for the three use cases described above, and the required transfer times corresponding to all combinations of the "song" and "favorites" media types with exchange rates of 10, 5, and 0.5 Mbps. For the simulation, the user was taken as moving at the fixed speed given in the table below for each use case. For each use case, the peers were placed at random according to a Poisson point process in the plane with the average density given in the table and were assigned speeds uniformly at random between the minimum and maximum speeds given in table 2. In the 5th Avenue case, directions were chosen as parallel or anti-parallel to the user with equal probability. In the other two cases, directions were chosen uniformly at random. The effective range of the connection was taken to be 9.14 meters (30 feet). The simulated duration of the experiments was 1000 seconds.

Table 2. Simulation use cases and characteristics

Use Case	User Speed	Peer Density	Peer Speed	
			minimum	maximum
5th Ave.	0.91 m/s	1.0/m^2	0.91 m/s	1.8 m/s
Campus green	0	0.25/m^2	0.25 m/s	2.0 m/s
Grand Central	0.91 m/s	3.0/m^2	1.8 m/s	1.8 m/s

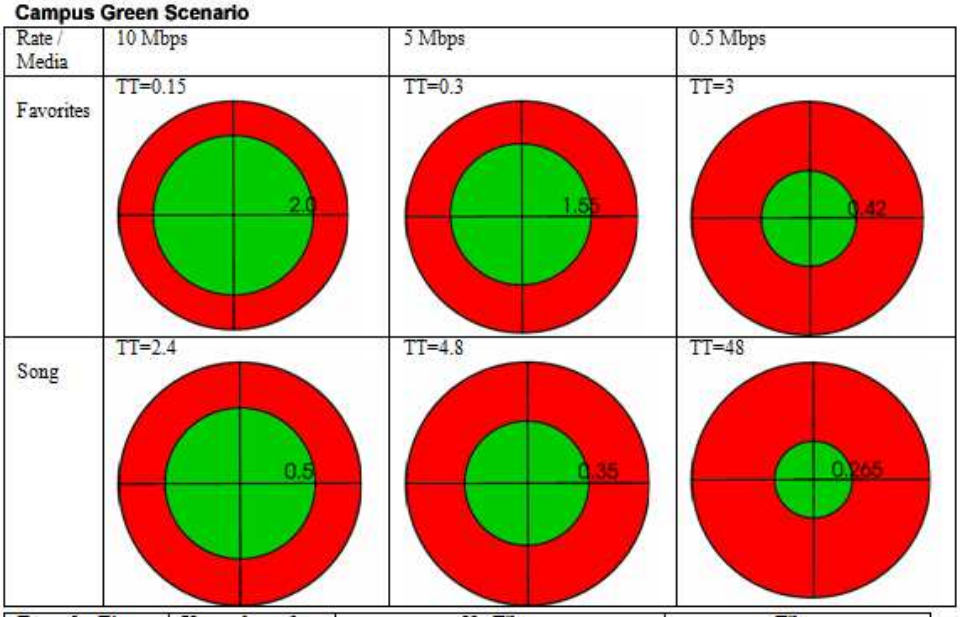
For each total transfer time we chose a sample filter by hand, as a user might. We tried to maximize the number of successful transfers, using as a rule of thumb the ideas that the expected rate of arrival of peers that pass the filter should be a small multiple of the rate at which complete information transfers can occur, and that we want to filter out the peers that are likely to stay in range for the least amount of time. No formal or analytical optimization was done. Any such optimization could only improve the performance of the filters beyond the promising results we report.

After generating a set of peers, we can measure how many of them are within range of the user during the duration of the experiment. We can also divide the duration of the experiment by the required duration of a successful transfer, giving the number of transfers we could achieve if we always had an available peer within range, who would wait within range until the transfer were complete. In the particular cases whose results are described here, the latter number is almost always smaller. The minimum of these two values gives an upper bound on the number of possible transfers, and a good benchmark to which to compare the actual numbers of successful transfers.

At time zero the user starts out free. During the experiment, whenever the user is free, she picks a peer at random from the set of unused peers that are within range and, if a filter is active, that pass the filter. If there are no unused peers within range, she waits until a peer comes within range, and then selects that peer. She attempts information transfer until either the required duration elapses, giving a success, or until the peer moves out of range, giving a failure. The peer is then marked used, and the user does not attempt to connect to that peer ever again. At the end of the experiment, any peer still in transfer is recorded neither as a success nor as a failure.

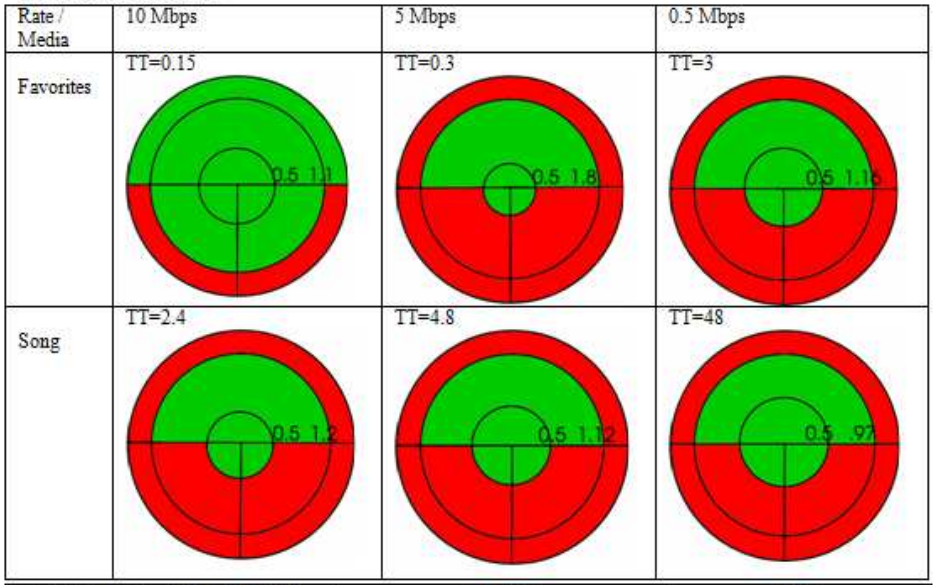
For each use case, we give the value of the bounds described above, and the numbers of successful and unsuccessful transfers both without and with filters.

Campus Green Scenario



Transfer Time	Upper bound	No Filter		Filter	
		successes	failures	successes	failures
0.15 s	5154	5115	3	5117	1
0.3 s	3333	3310	44	3318	15
2.4 s	416	378	77	410	19
3.0 s	333	301	62	326	13
4.8 s	208	177	67	191	13
48.0 s	20	1	70	8	14

5th Avenue Scenario



Transfer Time	Upper bound	No Filter		Filter	
		successes	failures	successes	failures
0.15 s	6666	6611	102	6648	38
0.3 s	3333	3271	111	3331	5
2.4 s	416	362	116	402	2
3.0 s	333	276	121	304	1
4.8 s	208	154	113	203	4
48.0 s	20	5	76	12	3

Figure 2. Filters and results: Sharing favorites and songs across 3 different data connections

Table 3. Effects of varying filters in Campus Green use case

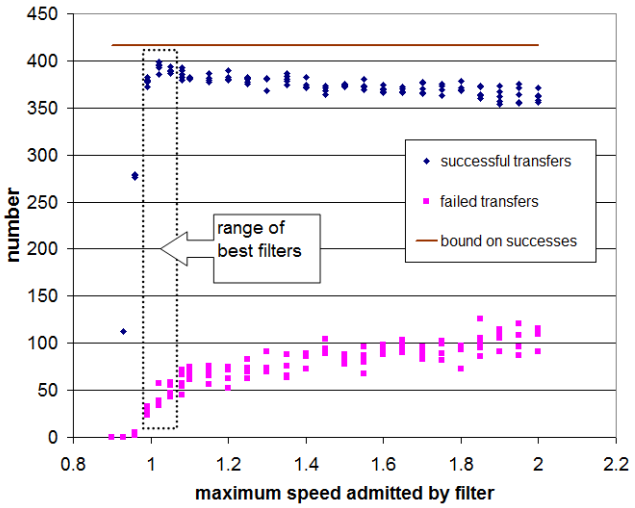
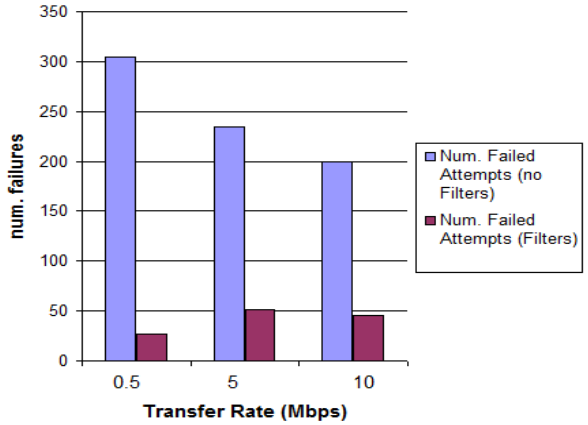


Table 4. Effects of filters across all use cases



3.1 Discussion

Figure 2 illustrates 5th Avenue (top) and Campus Green simulation results. We note significant reductions in sharing-sessions that fail (after starting, thereby wasting resources) when using geospatial filters. In the Grand Central use case (not shown), total failures without and with filters amounted to 7.1% and 0.43% resp. In the 5th Avenue and the Grand Central cases, the density of peers is high, and so we can tightly filter them to bring the number of successful transfers to the maximum possible, while greatly reducing the number of failed transfers. On average, failures were reduced by 91% in the 5th Avenue case and 93% in the Grand Central case by using filters as opposed to without filters. In the campus green case, the peers are sparser, but we still can increase the successful transfers and reduce the failed transfers, using filters. In this case, on average, failures were reduced by 75% by using filters as opposed to without filters.

In the campus green case, useful filters can be characterized by a single parameter α -- the maximum speed that will be admitted. Table 3 shows how the numbers of successes and failures vary with respect to α, for the 2.4 sec. transfer time. We take 5 trials per α value in order to handle the randomized choice, when the user becomes free, of the next peer from among the peers in range. At α = 2.0 m/s, all peers pass the filter, so the results are the same as without filters. As α decreases, failures decrease and successes increase, until about α = 1.02 m/s. At this point, failures and success both swiftly decrease as α decreases. Filters with α near 1.02 m/s can be seen as "good" filters. Table 4 summarizes the clear value of filters by aggregating across all use cases.

4. DESIGNING AND IMPLEMENTING THE TOOL ON A MOBILE HANDSET

Our simulation and analysis have shown that the ability to edit and manage personal geospatially-sensitive sharing policies on the device is advantageous. This section describes the MotionMaps tool, which allows exactly that. In effect, we propose a visual "language" with which geospatial policies are built and tested. The tool is intended for a user - either the device owner or an administrator. The key goals and design considerations were:

- Present the three independent variables (i.e., speed, direction, and binding preference) in a single visual notation.
- Clearly distinguish between information type (i.e., playlist, Internet favorites, components).
- Make use of the intuitive red-yellow-green "traffic-light" color scheme for indicating binding preference
- Allow immediate policy creation, saving, and testing; also allow easy changes to the angular and speed boundary settings found on the interface
- Compact design compatible with small color screens

We make several reasonable assumptions: (1) Velocity information is directly or indirectly accessible from the device or a service provider and, given this, directions and speeds can be computed for any two peers, (2) Binding is an opt-in feature and user privacy is guarded, (3) The device's screen must display color at (least) 160x160 pixels, (4) the device is, for the most part, on-the-go, not stationary.

4.1 The MotionMap Tool

The MotionMap visual notation is annotated in Figure 3 and is the main visual metaphor with which device users design sharing policies. The notation is meant to be interpreted with an orientation so that the user should take the upward Y-axis as pointing in the direction of travel; thus the relative bearing 90° is off perpendicular to the user's right (no matter which way she's traveling). The notation serves as a coordinate space in which other dynamically encountered mobile peers can be plotted and the color at that point on the MotionMap indicates the binding preference for the encountered peer[2]. Configurable concentric circles in the notation represent increasing speeds; the faster a peer is "passing" the user (or visa versa) the further out toward the edge it will be plotted on the MotionMap. The concentric regions are subdivided into regions by a set of angular lines corresponding to relative bearings. In the figure, the lines are at angles of 30°, 60°, (90°), 120°, 150°, (180°) in both positive and negative values, however these settings are configurable from the tool itself

[2] Though vaguely compass-like, the tool does not act like a compass for the user at bind time. The compass-like metaphor is used only at policy-authoring time; at encounter time, binding decisions follow the specified policies and are likely to happen invisibly to the user.

allowing, for example, alternate angular divisions such as -10° and 10°. In Figure 3, "region A" corresponds to the speeds between 0 and 10 (km/h) and the relative bearings of -30 to -120°. In creating binding preferences from scratch, the user begins with a "blank" MotionMap (white, as in Figure 3) and proceeds to color-in the regions. The meanings of the assignable region-colors are taken from the traffic-light semantic:

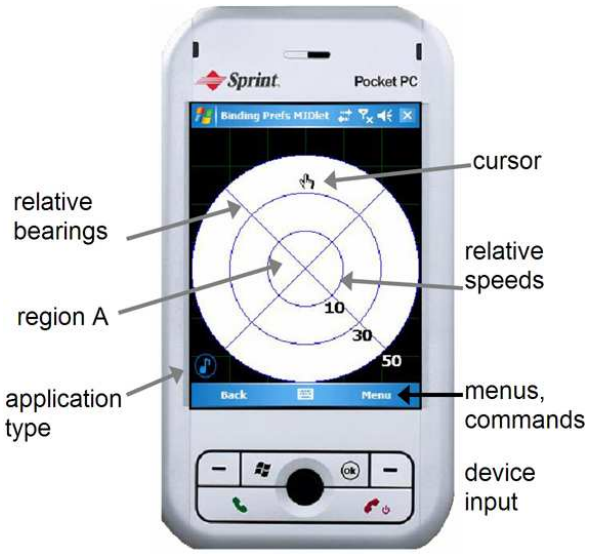


Figure 3. MotionMap notation on a "capable" device with a 240x320 color screen and a 5-way joystick

- Green - "bind to the candidate under these conditions"
- Red - "never bind to the candidate under these conditions"
- Yellow- "possibly bind" (e.g., probabilistically)
- White - no preference specified yet

To change the color of a region the user moves the cursor with the joystick to the region and clicks. Each subsequent click changes the color of the region, alternating through: white, green, yellow, red (then back to white) - multiple regions can also be selected and changed with a "change all" command. The visual notation captures bearings relative to the user in question, where 0° is co-incident with the bearing of that user. Thus, many MotionMaps will tend to be symmetric in the Y-axis since users will not likely want to distinguish peers approaching "from the left" versus "from the right" (though it *is* possible and more desirable in other commercial and military applications).

A small consistently-used icon on the editing screen's bottom left indicates the application type to which the policy is applicable (playlists, favorites, & components). Figure 4 shows user-created MotionMaps that reflect conservative and aggressive binding policies. In Figure 4 (left), the user paints green regions corresponding to peers traveling 15 km/h or less and traveling with a bearing that differs by no more than about 60°. By painting only in and around the co-incident angle the user ensures that binding will occur only with peers traveling the same or very similar direction. Yellow regions could allow for "random" binding to peers whose geospatial coordinates place it elsewhere in the MotionMap. Figure 4 (right) shows some of the menu commands available to the policy designer; at the right (partially obscured) the user specifies a "liberal" binding policy whose green binding area includes any peers with speeds in the range [0,20 km/h] at any relative bearing. The only binding situations this policy author rules out are when the candidate peer has a relative bearing in the range [120,240] and has speed greater than 15 km/h. "Moderate" binding policies would fall somewhere in-between these two MotionMaps while an unlikely or erroneous MotionMap might lack symmetry across the Y-axis.

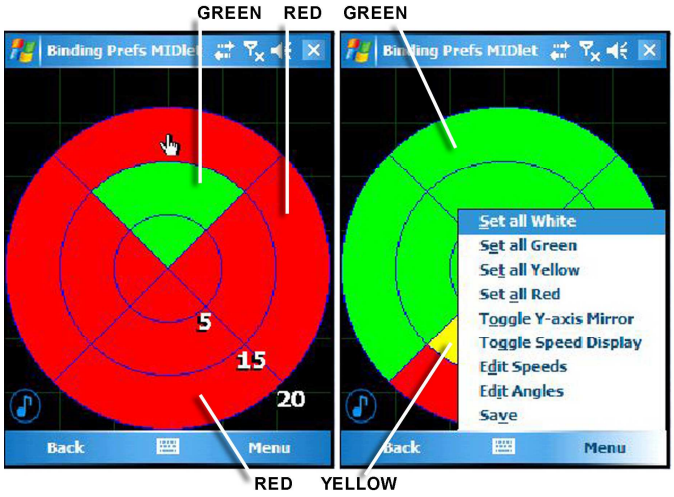


Figure 4. User-created playlist MotionMaps: (left) conservative, (right) liberal (colors are called-out for clarity)

4.2 Navigating through to Design

The tool's GUI asks the user to: 1) choose the relevant application, 2) choose either "design" or "test", 3) if "design" then choose the user's mobility level (e.g., foot, bicycle, car - i.e., slow, medium, fast), 4) choose to begin with a blank MotionMap, begin with a pre-populated one and edit it, or load a MotionMap from the storage card or a server. The user's choice of mobility level affects the speed boundaries that are drawn on the MotionMap canvas.

When a given MotionMap is ready to be employed by the user she must first save it and then activate it. The interface menus provide these features. Saving can be done onto local mobile storage or up onto a remote server if the device has an TCP/IP connection. As mentioned, we have also experimented with storing a rich representation of metadata using Semantic Web technologies [4].

4.3 Intuitive and Immediate Policy Testing

Once users are familiar with the MotionMap notation our tool makes the process of creating binding policies easy and repeatable. While the intuitive representation affords users a good understanding of the policy just by looking at it, sometimes just looking is not quite enough. An effective way to visualize and test policies before saving and activating them is to see them in simulated action. To this end, we have designed and integrated a mini-simulation screen on which MotionMap designers can test their sharing policies. They do so by choosing the "Test" choice instead of the "Design" choice during setup. The salient points of the subsequent test screen are:

- The "user" is represented at screen's center as an icon - (same screen coords that the MotionMap had)
- A small circle around the user symbolically represents the outer border of the user's wireless realm. That is, users

outside of this circle are not yet known to the user; as they enter this region they become binding candidates (as in real wireless discovery, such as 802.11b).

- Simulated mobile peers (initially red) move towards and across the user's realm. Upon entering the realm they change color according to their trajectory and speed and the sharing policy being tested - as they leave the realm (on the other side) they change back to red. Therefore a red peer indicates that no binding occurs with it - a yellow peer indicates "maybe" and a green peer indicates a "bind". These correspond precisely to the MotionMap policy designed in the previous step. The manner and number in which peers move at the user correspond to the user's mobility type

Figure 5 illustrates these concepts with an annotated diagram of the test screen (the added streak behind the peers illustrates their trajectories). The simulation is intended to be watched by the user who, in turn, as various simulated peers cross her path, gets an intuitive idea of the value of her MotionMap policy. In Figure 6 the first peer comes into the realm and changes to green indicating that the user would bind to this device if using the MotionMap policy being tested. At any time the user can add or remove peers into the simulation by nudging the device joystick right or left (resp.).

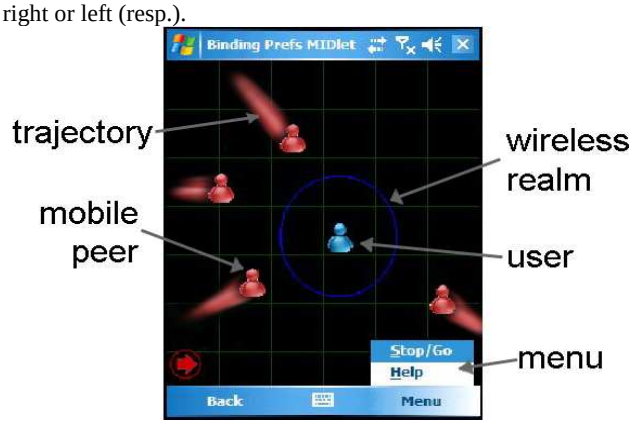


Figure 5. Annotated diagram of the test screen

Figure 7 shows a test after significantly more peers have been added. If not satisfied with the simulation results the user returns to the design screen where the MotionMap can be modified.

4.4 Technical Notes

The MotionMaps framework has been developed on a Java2 platform. Although today's mobile handsets have varying support for open (e.g., Java) and propriety (e.g., BREW) development middleware, Java's penetration into the market is great. In addition, the Java-based OSGi framework has a strong presence in the automotive market [7]. We have built the running prototype on a Windows Mobile 5 Sprint 6700 mobile handset as well as on Cingular HTC-based devices. The system is largely implemented as a Java MIDlet application, thus a MIDlet manager must be present on the MH Operating System. We have used the Tao Group's beta *Intent* MIDlet Manager on the mobile handsets but others exist, such as IBM's J9 MIDlet Manager; all provide fill-in support for MIDlets on mobile handsets that do not natively do so. The Sun Java Wireless Toolkit [9] and the open source cross-platform NetBeans IDE [10] have been used to build and test the project, first on software emulators and then on Connected Limited Device Configuration (CLDC) and Mobile Information Device Profile (MIDP) compatible devices. Software-wise the Policy Testing tool and interface makes use of Java objects called *MobilePlatforms* implementing *Runnable* (threads). These objects are created on demand and seeded with the geospatial attributes according to the test pattern; once seeded they follow their intended "trajectory" across the user's simulated wireless realm. Threads are destroyed as appropriate keypad buttons are pressed. The MotionMaps Policy editor makes extensive use of the Java2D [11] capabilities of the device. It manages a dynamic array of map segments and their attributes (e.g., their screen positions, their fill color) and uses methods such as g.fillArc( ) to draw segments.

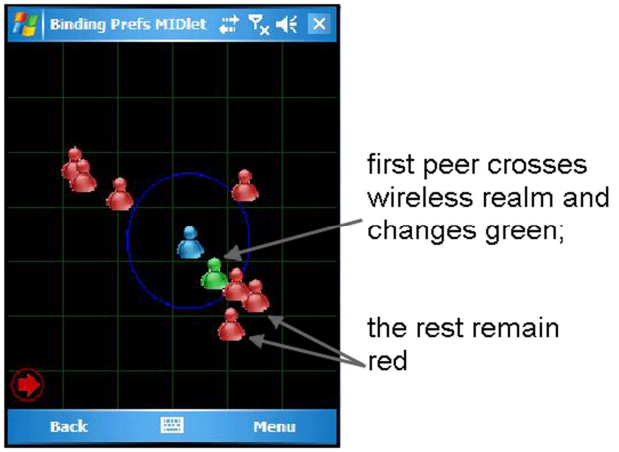


Figure 6. As simulated peers cross into the wireless realm of the user they change color according to the policy in test

Figure 7. More peers created by tapping on the button

5. RELATED WORK

Little previous work addresses mobile UI design and geospatial sharing preferences for mobile devices. Today's Instant Messaging (IM) tools use simple policies to block or enable incoming messages and location-enabled IM add-ons can offer notifications when any buddies are nearby [12]. Similarly, mobile location-based dating services send alerts when other compatible singles are nearby. In [13], a small representation of a bird on a browser toolbar is a dynamic visual cue and changes color (from green, to yellow, to red) according to how closely a web site's and user's privacy policies correspond. That users can grasp compass-like metaphors is known: in Japan, a deployed wireless service by KDDI uses a compass metaphor to "point" users to interesting locations and a different system successfully uses a (similar) metal-detector metaphor for location-sensitive services [14]. Microsoft Research's Scope uses a compact circular radar-like interface to display notifications to computer users [15]. However, these and others do not use nor require rich geospatial policies nor testing, and visual edit and test tools are not proposed. [8] and others survey the wide variety of user mobility models; in comparison, our work uses straightforward mobility models and constrains them to three distinct use-cases, each of which has particular geospatial characteristics. Wireless positioning technologies such have used accelerometers and gyroscopes to gage direction and speed in order to better track mobile wireless users [16]. Research into Mobile Ad Hoc Networks (MANET) and mobile peer-to-peer address related issues to those described here but also take into account Layer 2 and 3 concerns. For example, [17] describes how to exploit both information age and its position relative to where it was created in order to disseminate more effectively.

6. CONCLUSIONS

Through simulation this work has put a pragmatic explanation point on the importance of geospatially sensitive policies for information sharing in mobile peers. In most interesting cases there is great value to eliminating from candidacy the set of peers who will not be in range long enough to allow a complete information exchange transaction. Through simulation and design this work has resulted in an intuitive visual notation – called MotionMaps - for expressing geospatially-sensitive policies on the mobile handset itself, and a way for mass-market users to visualize, test and tune these policies. When enabled in the middleware of the mobile handset the policies provide continuous and user-transparent green-light and red-light decisions to inter-peer bindings. This work has high relevancy in mass-market and commercial senses as new devices emerge that are specifically aimed at ad hoc media sharing. Ongoing and future work includes a) keystroke level modeling and user assessments of the visual tool, and b) by way of models enabling the system to "advise" the user on potential utility or ineffectiveness of the filter she has created.

7. REFERENCES

[1] A.Kobsa, "Privacy Enhanced Personalization", Comm. of the ACM, 50(8), pp.24-33, Aug., 2007

[2] S.Avancha, A.Joshi, and T.Finin, "Enhancing the Bluetooth Service Discovery Protocol", Technical report, University of Maryland Baltimore County, TR-CS-01-08, August 2001

[3] H.Ishikawa, Y.Ogata, K.Adachi, T.Nakajima, "Building Smart Appliance Integration Middleware on the OSGi Framework", *Proc. IEEE Int'l. Symp. on Object-Oriented Real-time Dist'd Computing*, Vienna, May 2004.

[4] B.Falchuk, D.Marples, "Ontology and Application to Improve Dynamic Bindings in Mobile Distributed Systems", *Proc. 2nd Int'l. IEEE Wireless Internet Conference* (WICON'06), Boston, 2006

[5] D.Famolari, P.Agrawal, "Architecture and performance of an embedded IP Bluetooth personal area network", *Proc. IEEE Int'l Conf. on Wireless Communications*, pp.75-79, India, 2000

[6] Anything but iPod, Microsoft Zune Review, http://www.anythingbutipod.com/archives/2007/04/microsoft-zune-review.php

[7] Vehicle Expert Group (VEG), OSGi Alliance, http://www2.osgi.org/VEG/HomePage

[8] T. Camp, J. Boleng, and V. Davies. "A Survey of Mobility Models for Ad Hoc Network Research", Wireless Comm. & Mobile Computing, 2(5), pp.483-502, 2002.

[9] Sun Java Wireless Toolkit for Connected Limited Device Configuration, http://java.sun.com/products /sjwtoolkit/

[10] Netbeans Integrated Development Environment, http://www.netbeans.org/

[11] Java2D API Specifications, http://java.sun.com/j2se/1.4.2/docs/guide/2d/spec.html

[12] L. Cranor, M.Arjula, P.Guduru, "Use of a P3P User Agent by Early Adopters", *Proc. of the ACM Workshop on Privacy in the Electronic Society*, Washington, 2002

[13] J.Koolwaaij et al, "Context Watcher - Sharing Context Information in Everyday Life", *Proc. IASTED Web Technology., Applications and Services*, Calgary, 2006

[14] Y.Takeuchi, M.sugimoto, "Intelligent City Guide with Metal Detector Interface", *Proc. ACM UIST*, Oct., 2006

[15] M.Dantzich, D.Robbins, E.Horvitz, M.Czerwinski, "Scope: Providing Awareness of Multiple Notifications at a Glance", *Proc. 6th Int'l. Working Conf. on Advanced Visual Interfaces* (AVI '02), Italy, 2002

[16] L.Fang, P.Antsaklis, L.Montestruque, M.McMickell, M.Lemmon, Y.Sun, H.Fan, I.Koutroulis, M.Haenggi, M.Xie, X.Xie, "Design of a Wireless Assisted Pedestrian Dead Reckoning System", IEEE Transactions on Instrumentation and Measurement, 54(6), Dec. 2005, pp.2342 - 2358

[17] Y.Luo, O.Wolfson, B.Xu, "Spatio-temporal Approach to Selective Data Dissemination in Mobile Peer-to-peer Networks", *Proc. Int'l. Conf. on Wireless and Mobile Comm. (ICWMC'07)*, March, Guadeloupe, 2007

[18] C.Ververidis, G.Polyzos, "Extended ZRP: a routing layer based service discovery protocol for mobile ad hoc networks" *Proc. 2nd, Second Int'l Conf. on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous'05*), San Diego, July 2005