

Shibuya Star (by Peter Pesti)

Shibuya Star

A prototype submission for the Forged-By-IMS Research Competition

Category: Young Adult/Teens

April 20, 2007

Atlanta, GA

Team #14 / Shibuya Star

Members: Peter Pesti

Executive Summary

“Shibuya Star” is a collaborative, mapping-enhanced, location-aware 3G wireless IMS software application for GPS-enabled PDAs, that provides the ultimate empowering technology for instant mobs - haphazardly formed groups of people, and teens in particular.

The concept of smart mobs is based on the vision that communication technology can allow people to coordinate and collaborate with never before seen efficiency, thus forming an information processing, “smart” group. We believe that 3G and IMS technologies hold the key to the realization of this vision. Our application augments the physical world of homes, roads, city streets and transportation systems with a virtual interaction layer. While the formation of smart mobs is currently enabled "offline from the real world" (eg. via online newsgroups), our combined instant messaging and push-to-talk solution for people near the same physical location provides a virtual channel to make group announcements and decisions on site and just in time, regardless of group size. Additional features like finding and locating physically formed groups by searching their recent communications further extends the capabilities of mobile users. Our application is also an enabler for 'tuangou' or team buying, and may also be used in emergency or on-site event coordination.

Shibuya Star (by Peter Pesti)

Our primary target group consists of teens and young adults (12-26 year olds), who are always on the move, craving for attention of peers, eager to form groups, make new friends and always on the lookout for interesting activities, while also being the fastest in the creative uptake of new technological possibilities. The project's name draws inspiration from the fashionable area around Shibuya Station in Tokyo, which is a favorite meeting place for hip Japanese teens.

Table of Contents

Executive Summary	2
Table of Contents	4
1. A new approach to mobile interaction: the agora	5
2. Business model	10
3. Architecture.....	12
3.1. The Shibuya Star client.....	12
3.2. The Shibuya Star server	15
3.3. Messaging	16
4. User interface	19
5. Use Cases	28
6. Limitations and Future work.....	31
References.....	33

1. A new approach to mobile interaction: the agora

Shibuya is one of the 23 special wards of Tokyo, Japan: a hip and fashionable area for young people, featuring a lively nightlife and numerous shopping opportunities. Near the Shibuya Station train station is the bronze statue of Hachiko the “faithful dog”, who returned to this place to meet his owner every afternoon even after his owner’s death, for 11 years. Today, the statue of Hachiko is a popular meeting spot for the “i-mode generation”. Photos of the constant hustle and bustle of the nearby Shibuya Crossing are the basis not only of people’s mental images of Tokyo, but of entire Japan.

The first chapter of Howard Rheingold’s “Smart Mobs” book [1], is titled “Shibuya Epiphany” (Figure 1). The observation of teens’ mobile phone-centric street culture at the Shibuya Crossing has lead Rheingold to formulate the concept of “smart mobs”: The pervasive availability of mobile communication increasingly enables people to self-organize, and act as a “smart” collective entity in ways that were not possible before. Text messaging has enjoyed an unforeseen level of popularity among teens, precisely because of its ability to ensure intimately private, pervasive, nearly instantaneous, proximity-independent coordination with peers. We believe that adding location-awareness and full instantaneity to the mix will allow new modes of usage of the already well established and popular text messaging mode of communications.



Figure 1: Howard Rheingold, the author of “Smart Mobs: The Next Social Revolution”, in Tokyo¹

On the Internet, a “forum” is a part of a website where people can exchange written messages, discussing a topic, answering a question, or following any thread of thoughts and ideas that the participating members desire. The word “forum” is of ancient Roman origin, signifying the public space in the middle of a city. Internet forums have also been used for “tuangou” (Chinese for “team buying”): using the virtual space of the forum, participants agree well ahead of time on a store location and exact time, where and when the people from the virtual forum appear in person, to receive discounts for large quantity purchases of the same product.

¹ From <http://www.links.net/vita/hlr/2001-tokyo/>

The word “agora” is of ancient Greek origin, and is defined as “a place of congregation, especially an ancient Greek marketplace.”², and also: “An agora (αγορά), translatable as marketplace, was an essential part of an ancient Greek polis or city-state. An agora acted as a marketplace and a forum to the citizens of the polis.”³ While “forum” is understood as *a virtual place for discussion, on the web*, we propose “agora” to signify *a virtual place for discussion, in the real world*. The very popular SMS or i-mode form of mobile text messaging is analogous to the Instant Messenger programs used on the net. Forums didn’t have a matching pair in the mobile world; agoras can fill this void as virtual forums attached to physical locations. As a “virtual place of discussion, in the real world”, an agora is a user-defined region of the physical space (eg. the Earth’s surface), that acts as a common bond for messages in the virtual space. In addition to defining the spatial parameters of a new agora, the agora may also have a defined topic (just like web forums). Mobile users can join and exit agoras at will, and send messages to the agora. Communication between agora members is instantaneous and persistent at the same time: agora members immediately receive others’ new messages, and the succession of the messages can also be reviewed later by others, who want to learn the course of the discussion. Agoras are location-oriented at the core, as they are meant for mobile users, therefore their utilization is expected to be of a different style than web forums (and hence the new term for the mobile world, since “location-based forum” brings connotations from the web world).

² Answers.com

³ Wikipedia.com

In the most basic approach, an agora is a potential source of information about a place: a user is able to observe the conversation that has happened in the past, which might shed some light upon issues that the user is facing, whether it is finding a hidden store, learning about the story of a monument, getting others' opinion on a restaurant, getting discount coupons, or any other scenario. However, all of these cases only include retrieving information that has been contributed by the community in the past, and are thus only make for slight advance over web forums.

At the next level, the instantaneous nature of agoras are also taken advantage of: the user is able not only to ask for some information, but coordinate physical action with other users, as is the case when a group of girls and a group of boys arranges to see a movie together while in a mall; when people share details of a potential bulk purchase deal with others nearby, in a real-time and spontaneous version of tuangou (team buying); when two backpacking American couples find each other in Xi'an, China and share their adventures and travel tips over a drink; when student on campus are notified and can immediately talk within the community of a serious incident; when firefighters advise office workers of the best way to escape a building on fire; when emergency officials guide evacuation efforts with localized real-time advice in a city struck by severe weather. This mode of usage of agoras is substantially

beyond that of web forums, as it enables localized fast information exchanges, and the creation of new small groups in the real world.

At the highest level of usage the historic location information of interaction in agoras is also utilized: physical interaction can be coordinated in a recurring way, based on past location-based interaction, thus enabling forging a community in the real world, as in the case of notifying people who were present at last week's political rally of a new rally; inviting your new friends from that grill party to see a movie with you; trying to find that girl you saw at the supermarket the other day, but who was gone before you could ask for her phone number; reconnecting with the people you met on your road trip at that trucker stop in Nevada, in Aspen on your ski trip, in Miami during spring break, or in Monaco last summer. The importance of privacy must be stressed: in all these cases past interaction based communication is established without the user having access to other users' interaction history: the user is only able to specify the context (time, location, keyword content) of the users he wishes to interact with, but the list of users matching the criteria are never made available, and thus the users' identities and past activities remain hidden. This mode of usage of agoras allows types of interaction which were never possible before.

2. Business model

We believe that open-minded business models are necessary for the success of IMS applications to avoid a fate like that of WAP, and to promote widespread use as the first priority.

The basic agora service should be offered free for a period of 6-9 months – enough to cover several seasons, and thus allow users to find and explore scenarios that are personally meaningful to them. With the expiration of the introductory period, the service can be offered on usage-based terms, where each posted message would cost approximately half of the price of a single SMS (ie. 5 cents), while viewing can remain free. The abundance of messages sent from users into agoras is critical for others' willingness to contribute, and thus a lower-than-SMS per-message price positions the system as “virtually free”, and can thus generate higher revenues thru the increased volume of usage. Allowing users to freely view posted content contributes to a “you can look but you can't touch” effect, constantly enticing users to get active by sending messages themselves – which they can then easily do with the per-message fee plan.

The pricing structure can be further developed by introducing “unlimited” posting plans offering 300 messages for \$5 and 2000 messages for \$10.

Bundled advanced features (such as push-to-talk voice chat with other users currently present in an agora) can act as even more potent profit

generators. The service is marketed as “hip” to a customer segment where social standing is of high importance; therefore advanced pay-features are expected to be popular once awareness and penetration of the basic service is sufficient.

There is also a possibility for advertising revenue. With the availability of a large body of location- and user-attached text messages, the hugely successful context-based advertising model could be adapted from the web to the mobile world, where context-awareness is even more important. Mobile advertising can be experimented with by offering a context-aware ad supported, but otherwise completely free subscription option in a few geographically confined pilot areas (esp. the Bay Area).

3. Architecture

3.1. *The Shibuya Star client*

Shibuya Star uses a broad range of technologies to provide its services (Figure 2). The client application, written in C#, is designed to run on PDAs running Windows Mobile 2003 or later, and equipped with the .NET Compact Framework (CF) 2.0 version. The current implementation was tested using the wifi Internet connection of a HP iPaq 6945 PDA. The application uses the redistributable DLLs of the GeoFrameworks GPS.NET toolkit [3] to access the built-in GPS of the iPaq. In addition to GPS, the user's location can also be detected automatically when connected to the Georgia Tech LAWN (Local Area Wireless Network), and using the GTLoc whereami wifi positioning system [7]. The satellite, aerial or map images displayed on the client's user interface are retrieved from the Microsoft TerraServer [5], using its SOAP webservice interface. TerraServer is also used for coordinate transformations. The Microsoft MapPoint SOAP webservice is used (with a developer license) to geocode specific addresses entered by the user. The satellite imagery navigation user interface was created by the author in the mGraffiti project [2].

TerraServer stores image data of the Earth at several pre-computed zoom levels. Image data can be accessed in 200 by 200 pixel tiles at all zoom levels. While TerraServer's interface is latitude-longitude coordinate oriented, its tiles

can be stitched together to provide a flat projection of the Earth's surface. TS uses the quite widespread and standard UTM NAD83 projection system to achieve this. The Universal Transverse Mercator (UTM) system breaks the surface of the Earth to 6° wide zones. A single projection from the sphere to the flat surface is used in each zone – as a result of the projection, the flattened zones are widest at the equator and narrowest at the northern and southern ends. Areas around the poles (84° North; 80° South) are not included in the zones, as projection error makes this impractical. The North American Datum of 1983 (NAD83) is one agreed-upon association of Zone 1 to a part of the Earth's surface. The coordinate system inside each zone uses the horizontal equator and the vertical central meridian as the “easting” and “northing” axes, respectively. As a result, both northing and easting coordinates can be negative. The unit of measurement is meters on both axes. The oval shape of the projected zones means that easting values might be valid or invalid, depending on the northing value. Image tiles are cached with a Most Recently Used (MRU) strategy on the client to reduce waiting delays. The utilized caching component combines a hash table (for fast lookup) and a linked list (for MRU replacement) [8].

Shibuya Star (by Peter Pesti)

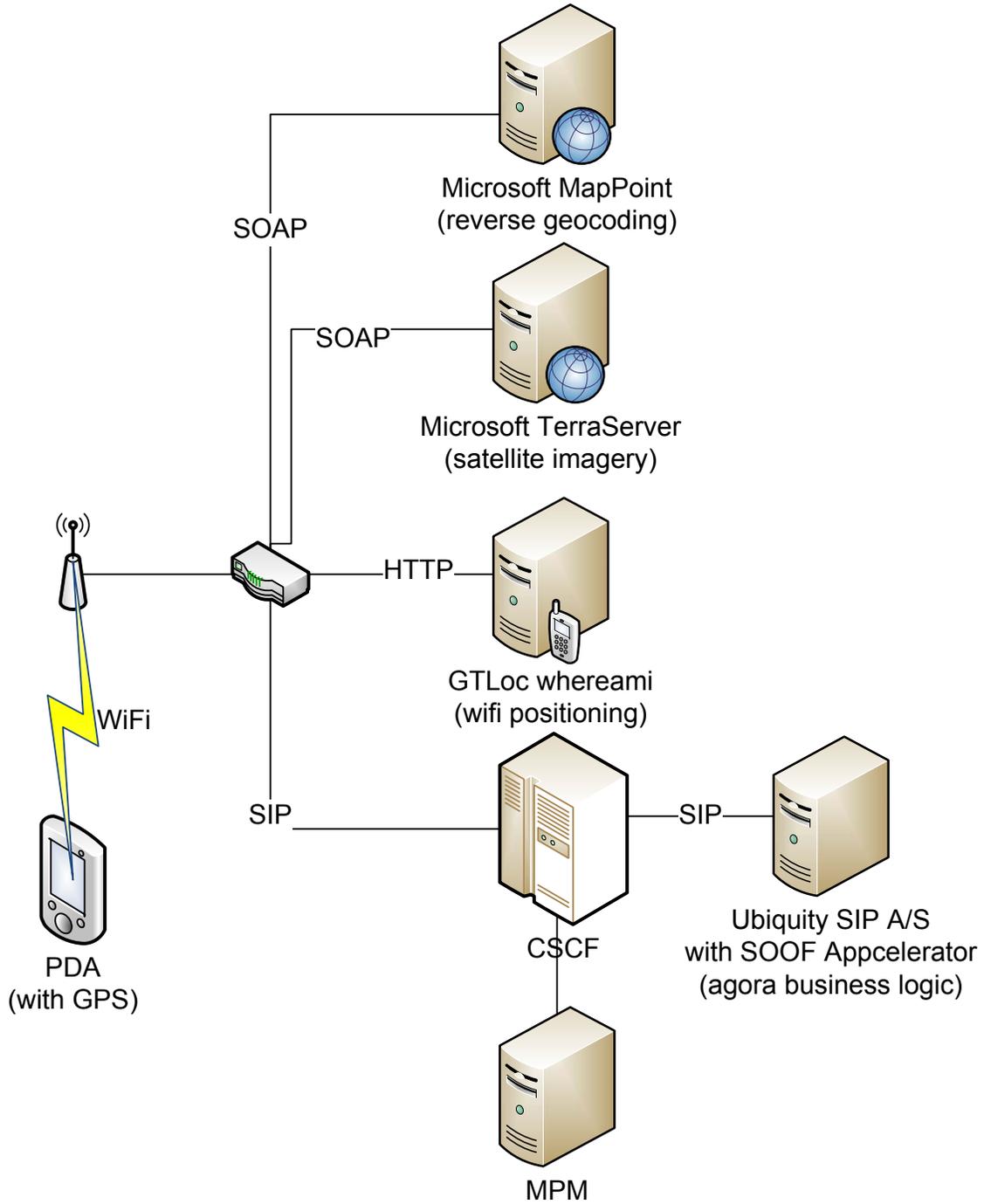


Figure 2: Simplified network architecture of the prototype implementation of Shibuya Star

3.2. The Shibuya Star server

The management of agoras and the passing of messages are handled by the Java servlets deployed on a Ubiquity Application Server with SOOF Appcelerator. Communication between the client and the A/S is done by SIP messages. On the client, the AgoraMessaging COM component was created with the use of the Siemens IMS WIN.NET SDK [4], to act as a messaging layer, composing and interpreting SIP messages. Server-side functionality is implemented with a two-layer architecture: each component is a SOOF (Service Oriented Object Framework) component. The handling of SIP messages is done by the AgoraMessaging FEB (Foundation Element Block), while the service logic is implemented in the ShibuyaStarService SEB (Service Logic Block), as seen on Figure 3.

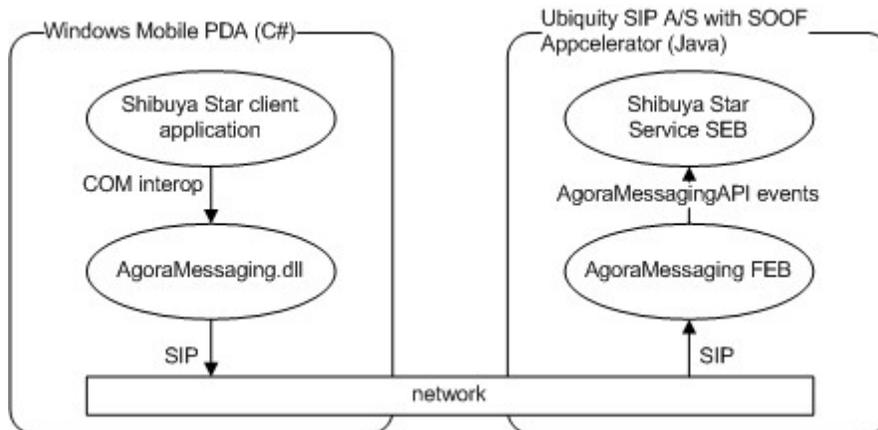


Figure 3: Core components of the Shibuya Star application and service

3.3. Messaging

Incoming SIP messages are forwarded to the `AgoraMessagingSipServlet` inside the `AgoraMessaging` FEB by the defined SIP Servlet mapping rules. After processing, the FEB fires the corresponding `SoofEvents` (defined in the `AgoraMessagingAPI`) into the A/S, which uses the defined `SOOF` Servlet mapping rules to forward these to the `ShibuyaStarService` SEB. The SEB handles all actions related to agoras, and the appropriate distribution of incoming messages to agora members. Ultimately, the server has five “methods”, which appear in some form SEB, to FEB, to SIP message content, and to `AgoraMessaging.dll`:

```
createAgora(UTMPoint center, double radius, String description)
```

Creates an agora with the given center, radius and description.

```
joinAgora(int agoraID, String userName)
```

The sending user joins the agora with the given ID, under the given alias. Each user can be in at most one agora at any time. Having joint an agora means that the user is able to send and receive messages to and from others in that particular agora.

```
exitAgora(int agoraID)
```

The sending user exits the agora with the given ID.

Shibuya Star (by Peter Pesti)

```
getAgoras(UTMPoint upperLeft, UTMPoint lowerRight)
```

Request a list of agoras that are at least partially inside the rectangle defined by the given upper left and lower right corner-points.

```
postMessage(int agoraID, String message)
```

The sending user posts the given message to the agora with the given ID.

The server is able to send three types of response messages:

```
sendMessageArrived(String message)
```

Notify the recipient that the given message has been sent to the agora that he is currently in.

```
sendAgoraCreated(int agoraID)
```

Notify the recipient that he has successfully created a new agora with the given ID.

```
sendAgoraList(ArrayList agoras)
```

Notify the recipient of a list of agoras that are of interest to him (eg. response to a getAgoras() call).

Table 1: Two example SIP MESSAGE payloads (with placeholders for call arguments)

<pre><?xml version='1.0' encoding='UTF-8'?> <AgoraMessaging> <type>createAgora</type> <center> <zone>{0}</zone> <x>{1}</x> <y>{2}</y> </center> <radius>{3}</radius> <description>{4}</description> </AgoraMessaging></pre>	<pre><?xml version='1.0' encoding='UTF-8'?> <AgoraMessaging> <type>postMessage</type> <agoraID>{0}</agoraID> <message>{1}</message> </AgoraMessaging></pre>
---	---

The method calls cross the network as SIP messages, created and interpreted by the AgoraMessaging.dll (using the Siemens IMS WIN.NET SDK) on the client, and the AgoraMessaging FEB on the server. These SIP messages are of type MESSAGE, and contain an XML doclet as their payload. The XML document defines the type of method call and the arguments of the call in plain text. Two example MESSAGE payloads are displayed in Table 1.

4. User interface

This chapter showcases the various user interfaces and interaction schemes of the Shibuya Star client. These screenshots were made from a Windows Mobile 2003 emulator with a rectangular screen form factor, matching that of the real HP iPaq 6945 PDA. Figure 4 shows the default satellite map navigator view: the satellite imagery shows downtown Atlanta.



Figure 4: The default map navigator view, showing two agoras

Four navigation buttons located at the border of the screen allow panning the map viewport north, south, east and west. The user can also pan the map by dragging it with the stylus, using a brisk motion. The two magnifier glass icons marked with "+" and "-" allow zooming in and out, respectively. The maximum zoom level has 0.25 meter/pixel (0.82 feet/pixel) resolution. Figure 4 also displays the two agoras that are within the viewport, shown by red circles.

Figure 5 shows the items in the “Go” menu: the user is able to locate himself using GPS (*Find me (GPS)*) or the GTLoc whereami campus wifi positioning system (*Find me (GTLoc)*): the map will be updated to display the area around the actual physical location, as determined by the selected service. Furthermore, the “Find address...” menu allows repositioning the map to any address inside the US (see Figure 14).

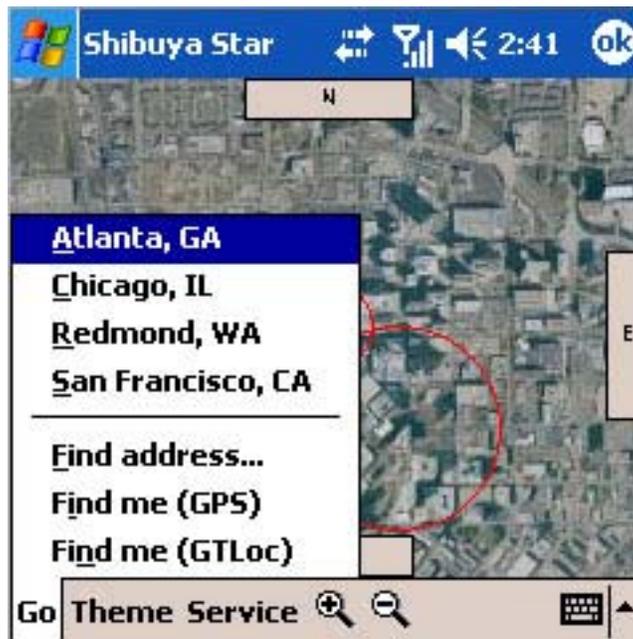


Figure 5: The "Go" menu allows fast repositioning

The menu also offers four predefined cities (Atlanta, Chicago, Redmond and San Francisco) for fast repositioning, principally for testing purposes. The most recently searched addresses can be displayed here in a future version, instead.

The “Theme” menu (Figure 6) allows changing the theme/layer of the imagery: the options are color satellite imagery, black and white aerial photograph imagery, or high resolution scanned USGS topographic maps. Generally, the satellite

imagery layer gives the most enjoyable experience. In Figure 6, the theme has been changed to aerial imagery.

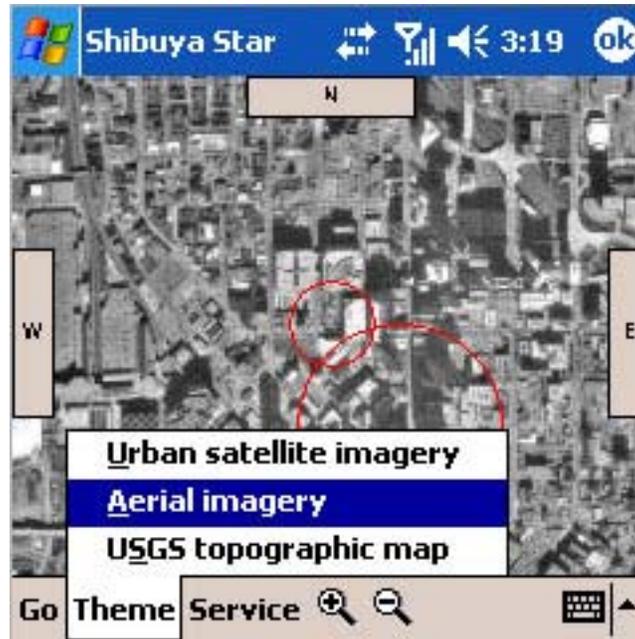


Figure 6: The "Theme" menu allows changing the imagery

Figure 7 shows the "Service" menu, providing controls that are mostly relevant for development: agoras can be turned off, and registration into the IMS network can be controlled.

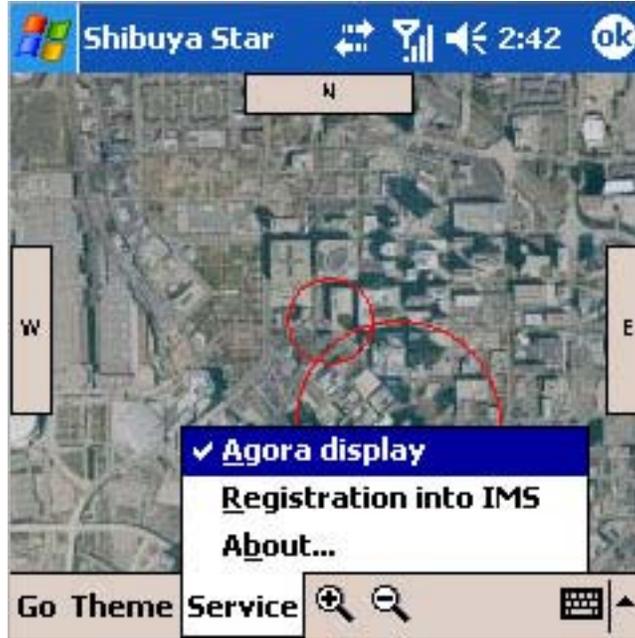


Figure 7: The "Service" menu allows advanced control

By tapping with the stylus on the screen, and keeping it held in place for 2-3 seconds (called a "tap-and-hold" operation), a context menu pops up (Figure 8). This menu allows re-centering the map on the selected location (*Center here*); zooming in and centering on the selected location (*Zoom in here*); creating a new agora centered at the selected location (*Create agora...*); or joining an agora that includes the selected location (*Join agora...*). An accidental tap-and-hold can be cancelled with the "Cancel" menu item.

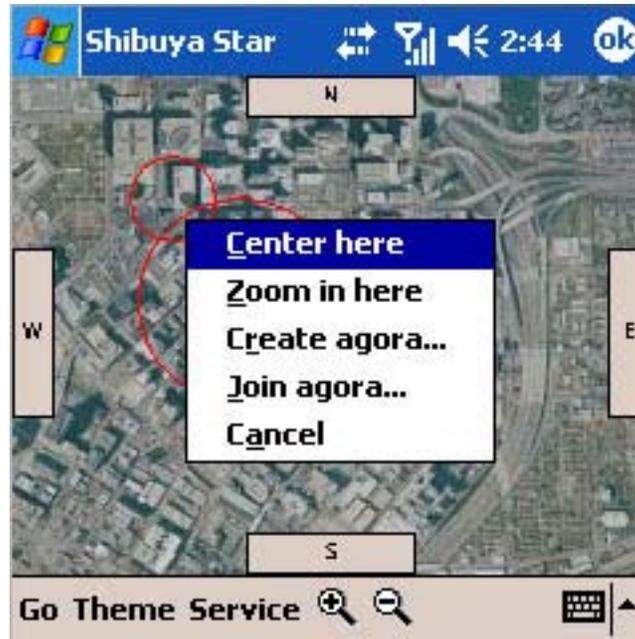


Figure 8: The popup menu, activated by tap-and-hold

Figure 9 shows the display after zooming in twice on the intersection of the two agoras with the “Center here” operation, and then panning west with the “W” button.



Figure 9: The view zoomed in on two agoras

Doing yet another tap-and-hold on the intersection of the two agoras, and selecting “Join agora...”, the list of agoras covering the selected location is displayed (Figure 10).

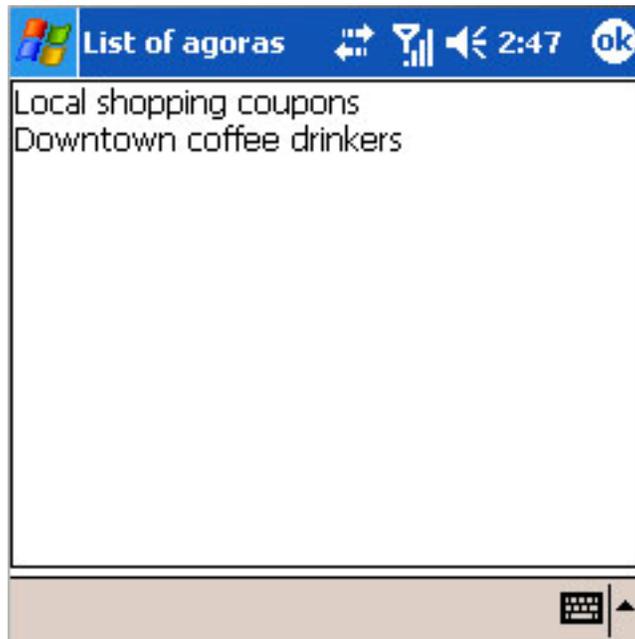


Figure 10: Selecting "Join agora..." from the popup menu reveals two agoras at the selected location

Selecting the “Downtown coffee drinkers” agora in the above screen brings the user to the agora talk interface (Figure 11). The title of the agora is displayed at the top. The user can enter his message in the box that reads “enter message here” (this message automatically disappears when the text box is selected), and then clicking on “Say!”. The messages from all users are displayed in the box below.



Figure 11: Selecting the agora with the topic "Downtown coffee drinkers" allows communication with other users in that agora

A user can not only talk in existing agoras, but can easily create a new agora. The process is initiated by marking the center of the new agora with tap-and-hold, and selecting "Create agora..." in the popup menu. After this, the user is in agora creation mode: a new tap-and-hold operation can be used to designate the border (and thus radius) of the agora. This modified interpretation of the tap-and-hold operation is also reflected by a modified popup menu (Figure 12).



Figure 12: After previously selecting the center of a new agora with the "Create agora..." menu item in the popup menu, a new tap-and-hold operation allows defining the boundary (radius) of the new agora, in this modified popup menu

After the center and boundary of the new agora have been defined with the tap-and-hold operations and the selection of the "Create agora..." menu items, the last step is to define the title/topic of the new agora, which can be done using a simple text entry form (Figure 13).

Finally, we must mention the form displayed by the "Go/Find address..." menu item (Figure 14), which allows address lookups even with partial information. At minimum, some part of the street name has to be entered, and at least one of the city, state, zip code field filled out. Clicking the "Go" button will result in the matching exact addresses being displayed in the box below. The user can then select the one that he had in mind, and be taken back to the map navigation display, where the area around the newly selected address will be shown.



Figure 13: The creation of a new agora is concluded by entering a title/topic for it

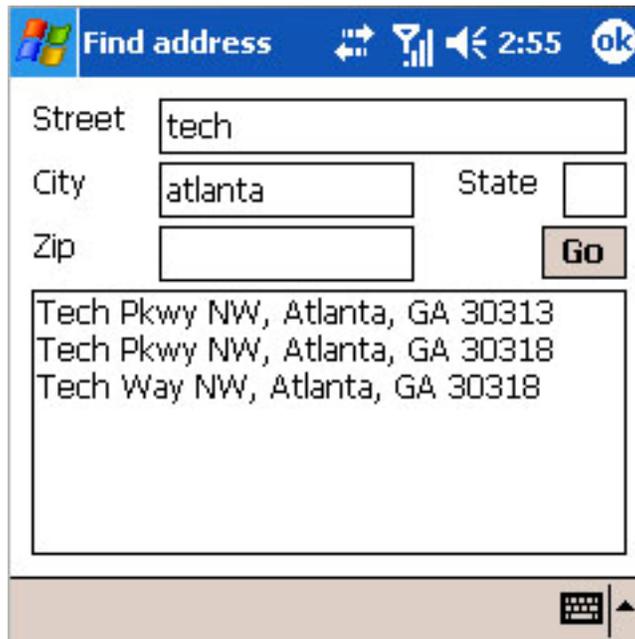


Figure 14: The address lookup interface allows locating addresses even when the address is only partially known: in this example, “tech” street in “atlanta” city returned three matches

5. Use Cases

1. Local discussion

1.1. Judy is sitting in a boring lecture. She brings up the Shibuya Star interface on her mobile phone and searches for conversations within the same room.

1.2. Using WiFi access point based location detection, the Shibuya Star server finds one discussion at Judy's immediate location.

1.3. Judy opens the local discussion and opts to see the profiles of participants who authorized this.

1.3.1. Judy sees that two of her friends have joined the virtual discussion in this classroom.

1.3.2. Judy sees that the new exchange student is also in the discussion, and he has a video profile.

1.3.2.1. Judy watched the video introduction of the new student.

1.3.3. Judy joins the discussion with some written comments about the lecture.

1.3.4. Judy receives and reads others' written comments in real time.

1.3.5. With the lecture over, Judy must rush to her next class. As soon as the current class ends, she uses push-to-talk to tell the new student that she liked his profile. Both run off in different directions, but they continue to chat for a minute.

2. Activity coordination

Shibuya Star (by Peter Pesti)

2.1. Frank was shopping in the mall alone. He sees a billboard for a good movie, but he doesn't want to go alone. Using Shibuya Star, he searches for virtual conversations inside the mall.

2.2. He finds a conversation, which he joins and asks if anybody is interested in this movie.

2.3. Two girls in the mall, who were looking for some shopping tips using Shibuya Star notice Frank's message and after checking his profile decide to join him.

2.4. Both of the girls use push-to-talk on their mobiles to chat in a group directly with Frank and arrange the meeting.

3. Team buying

3.1. Frank and Judy are strangers. They are both riding northbound on the interstate with their friends, unbeknownst to each other. Both of them use Shibuya Star to look for discussions on this section of the interstate.

3.2. Both of them find and join the local discussion and discover a promotion by a restaurant near exit 100, that offers a huge discount for groups of 6. Neither of the cars has enough friends in it, so Judy and Frank both send a message to the discussion looking for others to join them.

3.3. Frank and Judy discover that they had the exact same idea, and after checking each others profile, exchange a few written messages to assure the compatibility of the two groups.

Shibuya Star (by Peter Pesti)

3.4. After checking that the two teams would be happy together, they use push-to-talk to finalize the meeting, which really happens in 10 minutes, and the joined teams get the discount at the restaurant.

6. Limitations and Future work

The application is an early prototype that can be extended and developed in a number of ways.

The most obvious enhancement is in making the Shibuya Star service High Availability (HA), by persistently storing agora information and message history in the database attached to the A/S. The prototype simply stores all this information in memory, so everything is lost on a restart.

The unconstrained creation of agoras raises the problem of “agora-pollution”, ie. too many agoras cluttering the display, many of which are irrelevant to the user. This problem can be solved by allowing the user to filter the agoras presented to him, possibly by a combination of criteria, which can include attributes of the agora such as:

- the time of creation (historic view)
- the total number of messages (“where is lots of info?”)
- the number of messages in the last hour/day/week (“what’s hot?”)
- presence of keywords in the agora’s title
- presence of keywords in the messages

TerraServer’s large, 200 by 200 pixel size map image tiles cause sluggish map navigation because of the required remote server access and download

dealy. Client performance can be improved by pre-processing map tiles to better fit the small PDA screen. Smaller image tiles will mean shorter wait times when navigating in the map view. Furthermore, predictive caching could be used: pre-fetching of image tiles in neighboring zoom levels, or just outside the area currently visible on screen, for shorter wait times. Hosting all image tiles in the system seems to be an unviable approach because of the sheer volume of currently available satellite imagery, but providing a (non-photographic) map layer could indeed be solved.

Further enhancements in functionality would be the creation of user profiles, and the ability to send PMs (private messages), ie. make a private conversation instead of talking in the open public space of an agora.

In addition to messaging, communication between users could be made by voice: each agora would have its own push-to-talk (PTT) channel for live spoken chat. We must note, however, that this makes it difficult to process the content of the messages (eg. for filtering or extracting context information for advertising). Running a speech recognizer on voice chat could be possible sometime in the future with more robust recognition technologies, however. Applying a speech synthesizer to textual messages could then completely blend text and voice chats, with the option to participate in a conversation seamlessly in written or spoken form, regardless of the preferences of other participating users.

References

- [1] Howard Rheingold: *Smart Mobs: The Next Social Revolution*, Perseus Books Group, 2002
- [2] Peter Pesti, John Gibby: *mGraffiti: Scribbling the Virtual Earth*, 2005, <http://www.cc.gatech.edu/projects/dis/mGraffiti/>
- [3] GeoFrameworks GPS.NET, <http://www.geoframeworks.com/>
- [4] Siemens IMS WIN.NET SDK, <http://www.ims-developer-program.com/index.php?id=23>
- [5] Microsoft TerraServer, <http://teraserver.microsoft.com/>
- [6] Microsoft MapPoint, <http://www.microsoft.com/mappoint/>
- [7] GTLoc whereami, <http://www.rnoc.gatech.edu/whereami/>
- [8] Jim Wiese: Most Recently Used hashtable, <http://www.codeproject.com/csharp/mostrecentlyused.asp>