Hanging Messages: Using Context-Enhanced Messages for Just-In-Time Communication

Emily L. Chang - elchang@media.mit.edu
supervised by Professor Pattie Maes
MIT Media Lab - Software Agents Group


Abstract Many applications which have been introduced in the wireless arena have not capitalized on the unique advantages of mobile computing. Location based services, on the other hand, have started to capture widespread interest by using positional context to enhance functionality. Hanging Messages is a messaging system that uses both locational and temporal context, allowing users to specify time and place of message delivery. Additionally, agents reside on the client mobile devices, using the added message attributes in conjunction with constructed user profiles to effectively filter incoming messages.
# Contents

1 Introduction ........................................... 1

2 Background & Related Work ............................. 1
   2.1 Wireless Applications ................................ 1
   2.2 Location Awareness .................................... 2
      2.2.1 Commercial Products .............................. 2
      2.2.2 Related Research .................................. 3
   2.3 Location-Aware Messaging Systems .................. 4

3 Vision ..................................................... 5
   3.1 Focus .................................................. 6
      3.1.1 Message Categories ................................. 6
      3.1.2 Client-side Filtering ................................ 7
      3.1.3 Message Management ................................ 8
   3.2 Scenarios .............................................. 9
      3.2.1 Reminders .......................................... 9
      3.2.2 Requests ............................................ 9
      3.2.3 Promotions ......................................... 10
      3.2.4 Interesting Information ............................. 11

4 Design & Implementation ................................. 11
   4.1 System Architecture .................................... 11
   4.2 Technology ............................................ 12
   4.3 Implementation Issues ................................. 13

5 Project Plans ........................................... 14
   5.1 Research Stages ........................................ 14
   5.2 Resources .............................................. 14
   5.3 Timeline ............................................... 14
   5.4 Deliverables .......................................... 15

Notes ....................................................... 16
List of Figures

1  Hierarchy of Categories ......................................................... 6
2  Self Reminders ................................................................. 9
3  Requests from Others .......................................................... 10
4  Selected Promotions ............................................................ 10
5  Interesting Information ......................................................... 11
6  Overall System Architecture .................................................. 11
7  Client Software Architecture .................................................. 12
1 Introduction

As wireless communication technology has improved, the idea of “everywhere computing” has emerged; a world where PCs are replaced by various Internet-enabled devices that “sit in our pockets, our cars, on our desks, and in our homes.”

However, the currently available applications for mobile devices have not found favor with the average consumer. Companies have mostly attempted to adapt existing applications, such as web pages, to suit the new devices. This approach does not take advantage of the portable nature of the handheld computers, and is hindered by small size, low bandwidth, and slow processing speeds.

The solution is to develop applications that benefit from the unique features of “everywhere computing.” Hanging Messages is a system that uses place, time, and other attributes to make messaging more pertinent, since current messaging systems often result in information overload.

Message senders specify the location and time of message delivery in addition to the usual fields of an e-mail message. This reduces the number of messages actually delivered. Hanging Messages also builds on existing research by the Media Lab’s Software Agents Group, demonstrating how an agent can construct a user profile and use it to assist the user by filtering and sorting incoming messages. In this case the agent resides on each Hanging Messages client, processing incoming messages according to determined user preferences.

Hanging Messages exploits the mobility of Internet-capable handheld devices. The system requires that users carry their client devices everywhere, so that messages will be delivered at appropriate locations, and requires a continuous connection in order to retrieve messages in a timely fashion.

The greater functionality of the Hanging Messages system could not be accomplished without the advantages of portable computers and wireless communications.

In addition, the Hanging Messages system concentrates on maximizing the advantages of having additional contextual data. A client-side agent in this system has access to time, location, and category information. By using this data effectively in conjunction with the user’s profile, an intelligent agent can process incoming messages efficiently, and reduce the number of messages that are actually delivered to the user.

This paper discusses the Hanging Messages system and how it uses wireless devices to enhance messaging capabilities. The first section describes academic research and existing commercial products in the wireless arena, with an emphasis on location based applications. The vision for the Hanging Messages system is detailed in the subsequent section, along with scenarios in which it might be useful. Design and implementation issues are briefly considered, followed by an outline of plans for the completion of the project.

2 Background & Related Work

2.1 Wireless Applications

Commercial applications that have succeeded in the wireless arena include Short Messaging Services (SMS) in Europe, and NTT DoCoMo’s i-mode, in Japan. SMS focuses on services that take advantage of mobility, and are not limited by device technology. It does not allow Web browsing,
or full e-mail text reception, but does transmit e-mail alerts and other short messages which can be easily viewed in the small screen of a cellular phone. This functionality makes e-mail itself more timely, and does not attempt to replace it.

Users of i-mode, on the other hand, can send and receive full e-mail messages, and access information such as weather forecasts and sports scores. However, i-mode is "packet-switched," meaning i-mode devices can always be on, giving it an advantage over WAP. Another important difference is that space within the home in major Japanese cities is very limited, so PC users are far fewer than they are in the United States. Furthermore, i-mode uses compact HTML (cHTML), in contrast to Palm's "web clipping" technology, which requires that users run their documents through a proprietary compiler. Simple content creation has allowed ordinary consumers to help fuel an explosive growth of content.2

In the United States, use of wireless technology is mostly restricted to cellular phones. Blackberry Wireless E-mail has gained some popularity recently, with its “Always On, Always Connected” tagline, Blackberry Wireless devices have keyboards and thumb trackballs, and can be integrated with e-mail received on Microsoft Outlook.3 However, Blackberry only offers e-mail connectivity and organizer functions.

The Palm VII series is representative of most other American wireless handhelds. In addition to e-mail connectivity, Palm offers web clipping, which uses a query and response model to retrieve data. Web clippings are generally generated by CGI scripts in response to a query, but can also be static pages. However, Palm requires web clippings to be compiled through their site by their “Palm Query Application Server” which uses an HTML document to produce a .pqa file.4 This reduces the number of users who are likely to write their own clippings, and thus the total amount of content accessible through a Palm VII.

2.2 Location Awareness

One important feature of wireless devices is that they are portable, and can follow the user everywhere. SMS takes advantage of this by alerting the user to new e-mail regardless of whether she has access to a PC. Another way to capitalize on this mobility is to use locational context to improve the behavior of an existing application.

2.2.1 Commercial Products

Location based services are being seen by many wireless service providers as the “killer application” for wireless Internet, because they are helpful to mobile users. Several offerings have emerged from Europe in the past year, as well as a few from Japan and the United States.

In Germany, CT Motion and T-Mobil have teamed together to allow companies to monitor their employees, and intelligently send them relevant information based on location.5 Similarly, the Israeli phone service company Orange has a mobile workforce application that “offers companies the ability to monitor the movements of their workforce throughout the country to an accuracy of 100 metres.”6

Saverfone, based in the United Kingdom, allows users to find nearby promotions through WAP phones.7 In the United States, Phone.com's Mobile Location Server has allowed several companies
to venture into the location-based services arena. Phone.com has partnered with Amazon.com to develop m-Commerce applications, DoubleClick for location-based advertising, and In-Fusio for location-aware gaming, and Go2Systems for "community and content applications."

On the server side, Phone.com has teamed with SnapTrack, the leading wireless location server software developer.

InfoMove has taken a different approach, focusing on drivers rather than individuals moving by foot. Their services include "real time personalized traffic updates, real-time vehicle monitoring, profiled location-based advertising, [and] emergency assistance and concierge services."

Vert, located in Boston, has designed "iT pulver" which are geographically aware advertising units, that vary the advertisements based on location and time, in order to better target consumers. Location is determined using GPS, and the advertisements are delivered to the iT pulver units through a wireless network.

Most other services are of the "find-the-nearest" variety, allowing customers to look for the nearest retailers, restaurants and other services. Finland’s Sonera and Swiss operator diAx provide this service, as do various American service providers. Vicinity’s BrandFinder is a popular choice, provided by Verizon, OmniSky, AT&T Digital PocketNet, Palm VII and Nextel Online. BeVocal adds a voice recognition twist, allowing customers to speak the name of a business, to find the nearest office.

2.2.2 Related Research

Some applications of location awareness aim to optimize existing systems. Location-aware scheduling is one example; researchers at the University of Southern California have developed "locron", a cron-based scheduling system that uses location as an attribute. This system attempts to determine the optimal time to perform tasks that are battery draining, network bandwidth intensive, or otherwise resource dependent. For example, if the current location is not home or work, the user is most likely using a laptop, and should not back up his hard drive at that time.

Research on the device level that uses location information includes hardware that uses location to determine which kinds of data it will receive. If a device moves from Europe to the United States it should default to expecting a different format of data.

The Active Badge system developed in England by Olivetti Research Ltd. has been a popular concept with large corporations who wish to monitor their employees. Essentially, each employee wears an "active badge" which transmits the employee’s location every few seconds. It can also function as a entry card, and be integrated into security systems.

At Rutgers University, researchers have developed a system for marking web sites with specific locations, and subsequently retrieving those sites that are nearest in location, or accessing pages whose contents depend on user location. This system is intended for use with mobile clients, and indeed location-based content may improve the wireless web experience significantly.

In the early 1990’s, Lamming and Flynn of the Rank Xerox Research Center developed a system called Forget-Me-Not. This system used a PDA to capture data along with contextual information such as where the user is and who they are with. The information is stored in a database and can be queried by context such as location.
Several projects at the Media Lab have involved location based services.

In Wearable Computing, a project entitled "Memory Glasses" served as a short-term memory aid and reminder system. Location, time, and user activity were all considered when triggering reminders. This is similar to the Rhodes' Remembrance Agent, which in its wearable form, uses location as a context for suggesting reminders.

Impulse is an ongoing project in the Software Agents Group that combines e-commerce with "brick-and-mortar" commerce by using agents on wireless mobile devices to negotiate purchases with retailers near the user's location. A typical scenario involves a consumer who passes a shop which carries a product that she wants. The handheld alerts the user, who proceeds to examine the product at the store itself. Price comparison or other information can be accessed through the handheld, before making a decision to purchase. Various implementations of the Impulse scenario deal with different uses of location information.

Wherehoo, developed by Jim Youll, is a search engine optimized for location-specific searches, used as an infrastructure element in Impulse implementations. Periscope, a related project, includes a telescope-like device that feeds direction and distance parameters to Wherehoo, and retrieves data placed at the specified location.

2.3 Location-Aware Messaging Systems

Another interesting application of location awareness is messaging. Current messaging systems only allow the sender to specify the recipient(s) of a message. However, often messages are strongly context-sensitive. For example, a message might be appropriate only if delivered during a certain time interval, or most useful if delivered while the recipient is at a certain location. In both cases, using additional information about the recipient's context allows the delivery system to filter the messages. The result is that those messages which are delivered are of increased relevancy.

More recently, members of the Media Lab's Wearable Computing group completed a project called "Locust Swarm" which included the development of a location-based messaging system, which had a sophisticated location detection scheme that could determine location within a building. The Locust Swarm emphasized privacy by allowing the user to control how much information is shared with others.

Another implementation of location-based messaging is "comMotion," a project developed by the Media Lab's Speech Interface Group. This system learns relevant locations by observing user movement with Global Positioning System (GPS) sensing, and then associates "to do" lists with these locations. Users can also send each other reminders, and depending on privacy settings, query locations of other users. comMotion focuses on speech as an interface for managing reminders, so users can perform tasks while in transit. To ensure that user locations are not transparent to the server, messages are transmitted to the recipient as soon as possible, and retrieved when the client machine determines a positional match.

"GeoNotes" is a similar message-based system developed by the Social Mobile Computing Group at the Swedish Institute of Computer Science. Users of GeoNotes can annotate physical locations with "notes" which can be accessed and commented on by other users of the system, when they pass by that location. An interesting feature of GeoNotes is the push/pull model, where users can configure their mobile devices to one of three modes.
• push - User will be alerted when she passes by a highly ranked GeoNote, which will pop up on her device display

• push/pull - User can browse the nearby area for GeoNotes

• pull - User receives notes only through active searching, e.g., all notes placed by a particular person

The filtering done by GeoNotes focuses on usage, for example, notes can be categorized as “most read” or “most commented” and processed accordingly.25

3 Vision

The Hanging Messages system concentrates on adding context to messages to increase their relevancy. Increasing the number of required attributes on the sender’s side allows an agent on the recipient’s side to filter messages more intelligently, and allows the sender to incorporate his judgment into determining delivery context. However, the system is not a substitute for e-mail, it merely allows users an alternative channel through which to send time or location sensitive messages.

An agent is an autonomous program that can oversee a task on behalf of the user.26 The continuous, adaptive nature of an agent makes it the ideal solution for a task such as real-time message filtering. In the Hanging Messages system, all users carry handheld clients. The agents reside on the client devices, and monitor the user’s location, query the message server at appropriate intervals for messages, and process incoming messages, deciding whether or not to present them to the user.

As in comMotion, the client-side agents can learn locations that are meaningful to specific users, and develop theories for filtering rules, that can be confirmed by the user. Theories are proposed when a specific pattern of behavior is observed over a period of time. For example, if the user’s sleep schedule is such that she never wakes up before 9am on weekdays, and 11am on weekends, the agent can observe this, and theorize a rule. Because the agent asks for confirmation, the user can verify that she never wants to receive messages that come before 9am, but reply to the agent that she does want to receive messages before 11am on weekends. These rules collectively form a user profile that can, but need not be edited explicitly by the user.

Messages can be filtered by the agent into three types: active, passive, and ignored. Active messages will alert the user when they are delivered, while passive messages are not read unless the user explicitly queries a location for existing messages. Ignored messages are never seen by the user, but are kept so that they can be re-filtered if the user profile changes.

The hypothesis is that messages that are actually delivered will be more useful to the recipient, since the agent has performed some processing, but the user contributes to the behavior of the agent, and has ultimate control over her profile. In addition, the sender has submitted his input by choosing the time and place of delivery.

The core of Hanging Messages consists of a system that associates messages with specific locations and time periods. A sender composes a message, specifies the recipient(s), a location, a delivery time and an expiration time, and sends the message to the server. If an intended recipient comes within the designated location during the specified time period, her client device will receive the message.
The mapping of data to a geographical location is not a novel concept. Wherehoo27, implements this association, as does comMotion, which deals more specifically with messages. Hanging Messages focuses on using this location association to explore new commercial applications and client-side filtering ideas made possible by the added context.

### 3.1 Focus

The following sections describe important features of the Hanging Messages system beyond basic messaging capabilities. These properties are especially relevant to the focus of the project.

#### 3.1.1 Message Categories

The Hanging Messages system allows a message to be sent without a specific recipient. In this case, the message will be delivered to all users of the system who pass by a specified location. Among other possibilities, this allows commercial users to locally broadcast advertisements or special promotions.

![Figure 1: Hierarchy of Categories](image)

However, to allow client devices to effectively filter unwanted messages, an undirected message is required to specify a category. Possible categories are dictated by the server, and communicated to the clients when changes are made. The result is better targeting, which benefits the commercial
user, because messages are only received by consumers who are interested in their information, as well as the consumer.

Figure 1 illustrates the initial list of categories that will be allowed. The categories are universal, and stored on the server.

Categories have different levels of specificity, with the levels arranged in a hierarchy. For example, a user’s agent might accept messages from the general category “Promotions” or alternatively, it could accept messages only from the sub-category “Food” but not “Entertainment” depending on her preferences, as seen in Figure 1. This flexibility is significant, since it allows much more effective filtering.

### 3.1.2 Client-side Filtering

By adding location, time, and occasionally category designation as required attributes for messages, messages are automatically filtered. However, these additional criteria also make it possible to better process incoming messages on the recipient’s side.

Currently, an employee who goes on vacation for a few weeks returns to an inbox full of messages, many of which may be outdated. If e-mail messages were tagged with an expiration date, as Hanging Messages are, obsolete messages would never be received at all, reducing information overload.

Similarly, location can be used to eliminate unnecessary messages. Many e-mail messages are sent to mailing lists, sometimes entire companies. If specifying location were an option, notifications about facilities, for example, would only be received by those employees who were on-site. Telecommuters, or those on business trips, would never see the messages at all, since these messages are completely irrelevant to them.

These examples demonstrate automatic filtering; the sender’s judgment is used to determine the locations and time periods to which a message will apply. Additionally, an agent on the client-side can perform filtering tasks. If a certain user never looks at his messages before ten o’clock in the morning, because he is always asleep, the agent can observe this behavior and theorize a new behavior rule. Having developed a theory, the agent can confirm it with the user with a simple yes/no question. With an intelligent agent residing on the client-side device, these subtle patterns can be observed and used to further reduce unwanted data.

The same tendencies may surface along locational divisions; users may never want to receive messages while in places such as classrooms, churches, courts, or other locations where it is inappropriate to break one’s attention. In order to handle these preferences, each user has multiple modes, each with an associated user profile. Known locations default to one of the modes, but the user can override the defaults. Users can also edit the number of modes, as well as the filtering rules that apply to them. The default locations modes and profiles are:

- **Home** - Personal and Informational messages are active, all other messages are passive.
- **Work** - Work Related and Urgent Personal messages are active, all other messages are passive.
- **Social** - Urgent Personal messages are active, all other messages are passive.
- **In Transit** - Urgent Personal and Urgent Work Related Messages are active, all other messages are passive.
• Uninterruptible - All messages are passive.

This system assumes that user profiles are generally consistent for a given location. A user who might wish to receive personal messages as soon as possible while at home will probably not want personal messages while in an “Uninterruptible” meeting or class. However, it also allows the user to change modes; for example, if a user works from home, he can switch to “work mode” while still physically at home.

In order to ensure that users will not be bothered by unsolicited and unwanted communication, messages with no specified recipient are automatically categorized as promotional, and messages from unknown recipients are automatically passive. Users can edit lists of known recipients, and additionally they can join “recognition groups” of users. This is useful in a community such as a workplace, where all employees of a company can belong to a group and thus become known to each other.

Senders of messages can specify categories when composing a message, but they can be overridden by the recipient’s preferences. For example, a recipient can state that all messages from a certain sender are Work Related → Administrative, even if the sender marks them as Work Related → Urgent. This management policy gives the recipient ultimate control over message filtering and delivery.

Users of Hanging Messages can edit their preferences incrementally, as they use the system, or can use a web interface if they wish to make many changes at once, or circumvent being asked by their agents. The web interface will allow addition of user modes, editing of the profiles associated with the modes, and editing of recognition groups.

### 3.1.3 Message Management

One interesting question involves the further management of messages once they have been delivered to a recipient. Some messages have multiple recipients, and need to be delivered to each one which enters the specified location. Others have no designated recipient at all, but must not be delivered to a single user multiple times.

The Hanging Messages system solves this problem with the following protocol:

• Messages are stored until they expire, regardless of delivery status.

• When a message is received by a user, the user can choose to keep the message “active” if she wants to continue to receive it. In this case the user will receive the message again if she returns to that location.

• Users will not receive the same message multiple times while remaining in a location; the client device will monitor movement to see if the user has left a location and returned, or whether he has continuously remained there.

• A default timeout period is used to account for tracking error, to make sure users will not re-receive messages if they stray slightly from the target location and quickly return within the boundary.

• A recipient list is associated with each message on the server, even if the message does not specify recipients. This list keeps track of which users have already received the message, as
well as their preferences for receiving the message again. A message will not be re-delivered unless a user has chosen to mark it active.

This method gives more control to the user, and since messages are not active by default, most users will receive a message only once. The only case in which a message will be delivered to the same user again is if the user has decided that it would be useful to see again. Even promotional messages function the same way, making this system at least as efficient at reducing traffic as regular e-mail messaging.

3.2 Scenarios

This section describes several scenarios in which the Hanging Messages system is particularly useful.

3.2.1 Reminders

Users can send themselves reminders that will alert them at appropriate times and places. For instance, if the user is a student who often forgets to turn in paperwork on campus, she can send herself a message the night before, when she remembers:

![Reminder](image)

Figure 2: Self Reminders

When he walks by this location the next day, he will receive an alert. He will be reminded of his errand while he is in the vicinity, and it is convenient to carry it out.

3.2.2 Requests

Other people can send messages to users with location-specific requests. For example, if a user is working on a group project, and needs a specific book, she can send a message to all members of her group in the vicinity of the bookstore:
A group member who passes by and does purchase the book can then reply with a message to the same location stating that he has fulfilled this request, to prevent repeat actions.

3.2.3 Promotions

One of the more interesting features of Hanging Messages is that recipients need not be specified. This allows companies to send promotions to any users who come to a certain location:

The category system, previously discussed, prevents Hanging Messages users from being deluged with commercial messages. In fact, users can choose not to receive messages from any category. However, because the categories are flexible and precise, and most people do appreciate some types of promotions, it is anticipated that most users will not decline all categories of untargeted messages.
3.2.4 Interesting Information

Another benefit of allowing unspecified recipients is that location based news or other information can be posted for general interest purposes. One example might be at a local construction site:

![Message](image)

Figure 5: Interesting Information

In this example, the user has categorized messages coming from MIT News as Informational, so although there is no recipient it is not marked Promotional. However, the user is “In Transit” so the message is passive. Perhaps she is waiting at a stop light and has queried her agent for passive messages in the area.

4 Design & Implementation

4.1 System Architecture

The server and client software have been extensively modularized by functionality. This section explains the division of labor within each application, and illustrates their interaction with diagrams. Figure 5 illustrates the overall system architecture.

![Diagram](image)

Figure 6: Overall System Architecture
The major functionality of the server is implemented by the Message Manager, which receives incoming messages and stores them in the Wherehoo server. It also processes incoming requests for messages, keeps track of which users have received which messages, and cleans up the tracking mechanism when a message expires in Wherehoo.

The client-side architecture is detailed in Figure 6. The basic functions of the client are Message Retrieval and Message Processing. In order to receive messages in a timely fashion, the Message Retrieval module continuously queries the Hanging Messages server for appropriate messages. The Message Processing unit is in charge of filtering retrieved messages, using the user profile to determine which messages should be brought to the user’s attention.

![Hanging Messages Client](image)

**Hanging Messages Client**

Figure 7: Client Software Architecture

Location tracking is done by the GPS Module, which transfers its data to the rest of the client system through the Location Tracker. The Location Learning and Location Translation modules have been added for convenience, to allow the user to refer to locations through personalized names. Specifically, the Location Learning module tracks the movement of the user and allows her to name locations that she frequents. When the user uses these nicknames when composing an outgoing message, they are translated by the Location Translation unit before a message is relayed to the server.

### 4.2 Technology

The Hanging Messages server is a Java application that runs on a Linux box. The client software in this implementation resides on the Casio Cassiopeia e-125 running Windows CE. The development platform for the server application is Java 2, and client software development is done on eMbedded Visual C++. Location is determined by a GPS module on the client side, which is connected to a serial port on the e-125. Messages are stored on the Wherehoo server, which deals with persistence and expiration issues.
4.3 Implementation Issues

In order for the Hanging Messages system to function properly, clients must be able to quickly and accurately determine their locations. Some possible methods include triangulation using RF transmissions or cellular phone towers, use of Global Positioning System (GPS) technology, and various beacon-type systems. For this project, GPS was selected as the most appropriate, because it is accurate to 10 or 15 meters, and does not require the use of a cell phone or installation of hardware at relevant locations. Differential GPS (dGPS) is more accurate, to less than a meter; however, it is not well supported or widely used. Although GPS cannot be used to differentiate places within a single building, now that the government mandated 100 meter error has been removed from the GPS signal, it is adequate for the purposes of the Hanging Messages system.

Hanging Messages clients must constantly communicate in real-time with the server. One method of achieving this is to equip the client with a wireless modem, so that client-server interaction can be done over the Internet. The choice of client hardware is thus limited to those mobile devices which have at least two input/output ports; one to connect with the GPS module, and the other for wireless Internet functionality. The TRG Pro, developed by TRG, is one device that fits these requirements. The TRG Pro is essentially a Palm VII that is equipped with an extra CompactFlash slot and enhanced audio capability.26 However, the TRG shares the limited memory capability and restrictive development environment of other Palm OS devices.

The Compaq iPaq H3600 Handheld PC contains a StrongARM 206 MHz processor and is built with 32 MB of RAM, as well as ample audio functionality and a clear, color display.29 Windows CE is the primary operating system, which offers several development options for client software. An extension cradle adds a CompactFlash interface, and the form factor of the handheld is among the best in its class. The primary problem with the iPaq is availability; at this point iPaqs are back-ordered for three months or more, and the extension cradles are even harder to acquire. For this reason the iPaq was excluded from consideration as a hardware platform for the Hanging Messages client.

Like the iPaq, the Casio Cassiopeia e-125 comes loaded with Windows CE. With a 150 MHz processor, 32 MB of RAM, a built-in CompactFlash slot and a more brilliant display, the Casio is comparable in functionality to the iPaq H3600.30 Although the machine itself is slightly larger, the ready availability of the Cassiopeia e-125 made it the best choice for this implementation of Hanging Messages.

There are several options for Windows CE software development, the most popular being C, C++ and Visual Basic. Java was briefly explored as a possibility, but little documentation and the lack of a mature development platform for small devices suggested that a better supported language might be a better choice. C++ was decided upon as the best choice for this project, which is heavily modularized and suited to an object-oriented language.

One major concern in developing Hanging Messages was user privacy. Should the server be compromised, the secrecy of messages might not be able to be preserved, but at the least, client locations should not be disclosed. To achieve this result, clients track their own locations, and periodically query the server for messages existing at a given location. The server, after checking for a message match, discards the location information, which subsequently only exists on the user’s device. Location translation, location learning, and user profiling functions are also performed exclusively by the client agent, so that a user’s preferences are as secure if her client device is not compromised.
5 Project Plans

5.1 Research Stages

The initial implementation will consist of the server side, and a simple client, with only message sending and receiving capability. At this stage, the system will be functional, but will not filter messages for the user, and will require explicit location statements.

Subsequently, simple user profiling will be added, allowing incoming messages to be processed. However, the profile will be specified by the user, rather than determined by the client agent.

Next, location translation based on predefined nickname → location mappings will be added, followed by recognition groups. If time permits, preference and location learning functionality will be added to the client, to enhance the user experience of the system. Simulated locations, allowing the user to virtually exist at a certain location without actually being there, may also be implemented.

5.2 Resources

The hardware resources needed for completion of this project have already been obtained: one Cassiopeia e-125 and a computer capable of running the server application. A Java development platform, readily available from Sun’s Java site, has been installed, and eMbedded Visual C++ has already been acquired as well. Two Windows CE developer reference books, which are anticipated to cover the major functionality of the client application, have been bought to complement online documentation.

5.3 Timeline

- December 15th - implementation of server (Message Manager already implemented)
- January 11th - GPS → client → server data interfaces functional
- January 26th - implementation of UI
- February 9th - implementation of location modules
- February 23rd - finish basic implementation
- March 16th - implementation of client-side filtering
- March 30th - implementation of message features
- April 6th - finish overall implementation
- April 27th - preliminary draft of thesis
- May 11th - finish writing thesis
5.4 Deliverables

Among the products of this research effort will be the software itself; both the server application and the client-side agent. In addition, the written thesis paper will provide documentation regarding the development of the project, implementation issues, and usability commentary. If appropriate, short conference-style papers discussing interesting research issues may be submitted, as well.

Notes

1http://www.businessandfinance.ie/bandif/archive/20000601/pages/infoage1.html
2http://www.eurotechnology.com/imode/faq-gen.html
3http://blackberryonemain.com/
4http://www.webreference.com/dev/webclip/chipping.html
5http://www.ctmotion.com/content/PressDisplay2.asp?ID=8
6http://www.telecoms-mag.com/issues/200000/tdi/where_are_the.html
7http://www.wapland.com/art/350.html
9http://biz.yahoo.com/prnews/000927/1Phone_e.html
10http://www.infomove.com/pressreleases10.htm
11http://www.vert.net/inf0.html
13http://www.vicinity.com/vicinity/wireless/Bf_wireless.html
15http://www.it.kth.se/edu/PhD/LocationAware/Imed_BenDhaou_abstract.html
19http://www.media.mit.edu/ rich/memory/SIU00/Main.html
22http://wherewo.com/media.mit.edu/about/index.php


