Location Linked Information

by

Matthew William David Mankins
B.S. Mathematics
University of Miami, 1998

SUBMITTED TO THE PROGRAM IN MEDIA ARTS AND SCIENCES, SCHOOL OF ARCHITECTURE AND PLANNING, IN PARTIAL FULFILLMENT OF THE REQUIREMENTS OF THE DEGREE OF

MASTER OF SCIENCE IN MEDIA ARTS AND SCIENCES
AT THE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
SEPTEMBER 2003

Author: ______________________________

Matthew Mankins
Program in Media Arts and Sciences
August 8, 2003

Certified by: ______________________________

William J. Mitchell
Professor of Architecture and Media Arts and Sciences
Dean of the School of Architecture and Planning
Head of the Program in Media Arts and Sciences

Accepted by: ______________________________

Andrew B. Lippman
Chair, Departmental Committee for Graduate Students
Program in Media Arts and Sciences

LOCATION LINKED INFORMATION

Matthew Mankins

Submitted to the Program in Media Arts and Sciences, School of Architecture and Planning, on August 8, 2003, in partial fulfillment of the requirements for the degree of Master of Science in Media Arts and Sciences.

ABSTRACT

This work builds an infrastructure called Location Linked Information that offers a means to associate digital information with public, physical places. This connection creates a hybrid virtual/physical space, called glean space, that is owned, managed, and rated by the public, for the benefit of the populace. Initially embodied by an interactive, dynamic map viewed on a handheld computer, the system provides two functions for its urban users: 1) the retrieval of information about their surroundings, and 2) the optional annotation of location for communal benefit. Having the ability to link physical location with arbitrary information is an essential function to building immersive information environments and the smart city. Public computing systems such as Location Linked Information will enhance the urban experience, just as access to transportation dramatically altered the sensation and form of the city.

Thesis Title:	Location Linked Information
Author:	Matthew Mankins
Thesis Advisor:	William J. Mitchell, Professor of Media Arts & Sciences MIT Media Lab

Location Linked Information

Matthew William David Mankins
Master of Science Thesis
September 2003

SUBMITTED TO THE PROGRAM IN MEDIA ARTS AND SCIENCES,
SCHOOL OF ARCHITECTURE AND PLANNING,
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS OF THE DEGREE OF
MASTER OF SCIENCE IN MEDIA ARTS AND SCIENCES
AT THE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Thesis readers:

Advisor:

William J. Mitchell
Professor of Architecture and Media Arts and Sciences
Dean of the School of Architecture and Planning
Head of the Program in Media Arts and Sciences

Reader:

Tim Berners-Lee
Director World Wide Web Consortium
Senior Research Scientist
MIT Laboratory for Computer Science

Reader:

Michail Bletsas
Director of Computing
Research Scientist
Program in Media Arts and Sciences

ACKNOWLEDGEMENTS

Thanks to all that helped me through this: Bill Mitchell, Tim Berners-Lee, Michail Bletsas, Annie Berman, Mom, Dad, J.D., Parutron, Jennifer Beaudin, T.J. McLeish, Ted Selker, Kent Larson, Stephen Intille, Ken Wacks, Ralph Meijer, Peter Millard, Peter Saint-Andre, Rob Norris, Ryan Eatmon, Stephen Pendleton, Joe Hildebrand, Fritz Ganter, Darrin Eden, ryan mckinley, Stacie Slotnick, Frank Auer, Lee Templeton, Nicholas Yu, Dave Lampton, Leonid Fridman, Brad Harkavy, Amy van der Heil, Mark Avey, Sam Madden, Tim McNerney, Walter Bender, and the Smart Cities, Context Aware Computing, and House_n groups at the Media Lab.

Table of Contents

CHAPTER ONE: INTRODUCTION

'Electronic networking completes spatial networks of public spaces but it does not replace them. To believe the latter is a philosophical error of the same degree as believing that the wheel could replace the leg."
— Leon Krier

The Questions

What makes a good place to live? Can we maximize the urban experience by introducing a computational layer on top of our existing city infrastructure? How can we get the most out of our city? How long does it take to know what is going on in a location, or is it ever possible? Instead of putting the burden of information discovery on the city dweller, *what if the city could speak to you, giving you its hopes, aspirations, opinions, and past history*? Instead of humans giving up their bodies for virtual ones, what if urban space layered itself with virtual information? These are some of the questions that motivated this work.

Figure 1-1. This vision of the future city hoped the city would be more physically connected via various transportation methods, but is this still what we aspire for tomorrow's city?

Introduction

Set against the backdrop of an impending awakening of the city to computational technologies, this work builds an infrastructure called Location Linked Information ("LLI") which offers developers a means to associate digital information with public places. It is hypothesized that such digitally integrated cities provide richer interaction possibilities, allowing an increasingly mobile population to maintain a constant connection to the urban collective consciousness. Through the implementation of this technology, we consider the repercussions of digital computation in the city, culminating in a philosophical discussion of the friction inherent between virtual and physical realms in which we conclude that a refinement of the social contract to include a digital component is required.

Approaching the End of an Era: The Pre-Integrated City

The history of the city could be segmented into periods of "pre-technology," as in "the pre-automobile" city, or the "pre-walled city." We are living in the "pre-integrated" city, where outdoor computation and data access are private activities, carried out by individuals for their singular benefit. Whereas in the integrated city, the public sphere has incorporated the digital domain, providing a communal service through space-based knowledge sharing— just as aqueducts, roads, and the automobile benefited past urban states. This integrated vision of city form finds us sharing our streets with invisible, but undeniably present, digital information annotations that enhance the comprehensibility, and thus livability, of the city.

As we approach this new era of the city we are unprepared in both technical and social dimensions. Technically we are just beginning to explore the combination of physical and virtual, with movements such as Information Environments and Tangible Interfaces re-contextualizing interface design within physical space— yet there are many open problems in this area [32]. The good news is that the technical problems are being worked on, but the bad news is that there is little counterwork being done to adjust our social frameworks to accommodate these additions. Consequently, we regularly hear criticism, echoed in the dystopian novels *1984*, *We*, and *Brave New World*, of the horrors of future cities where technology brings over-order and an end to personal freedoms. Chapter five begins a discussion of a social component that technology designers need to be cognizant of as we enter this new era of *public computing*.

Figure 1-2. Technologies, such as the "wall" define periods of city history.

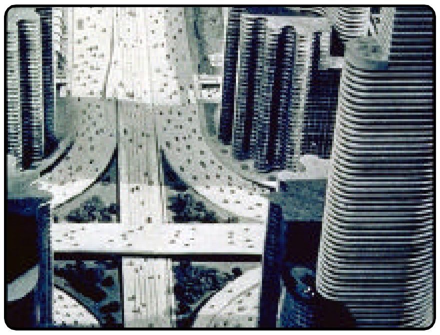


Figure 1-3. Futureworld exhibit by General Motors in the 1939 World's Fair imagined the future city would be defined by the latest technological addition of the time: the car.

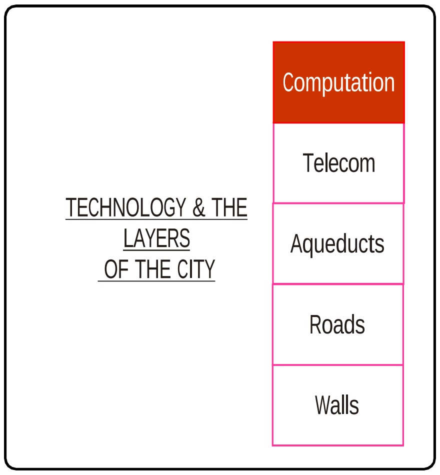


Figure 1-4. Computation as the latest layer of the city.

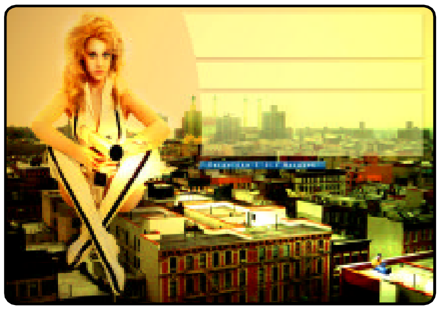


Figure 1-5. Projects like the EVil (East Village) Wireless Network of New York City are organically building a data layer on top of the existing city space, allowing the physical and virtual worlds to unite.

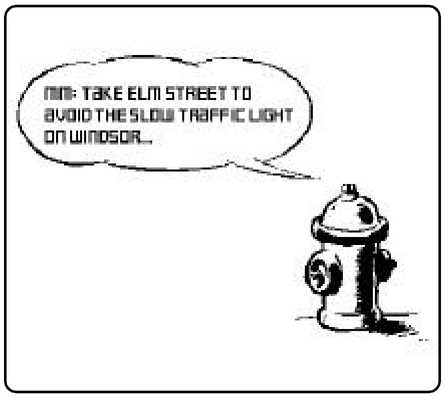

mm: TAKE ELM STREET TO AVOID THE SLOW TRAFFIC LIGHT ON WINDSOR...

Figure 1-6. Location Linked Information (LLI) introduces geographically-grounded virtual handles for retreiving and storing distributed information nuggets.

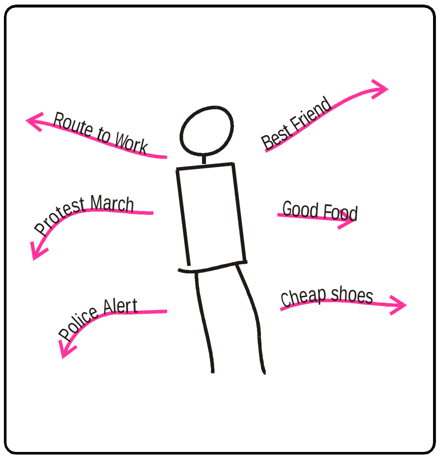


Figure 1-7. The LLI-enabled, hyper-aware citizen is in touch with his surroundings, able to “see more” through the aid of these digital tools.

Stepping Toward the Digital City: Location Linked Information

My technical contribution to building this new digital city, the Smart City, involves finding an answer to the question “how are we going to merge together physical space and virtual space *in the distributed public sphere*?” To this end, I designed LLI, a distributed infrastructure to support the access and creation of information “nuggets.” LLI couples a physical space/time moment with a distributed database containing information that describes that moment’s surroundings. This technology gives rise to a hybrid virtual/physical space that is accessed, managed, and rated by the public, for the benefit of the populace. Embodied by an interactive, dynamic map viewed on a handheld computer, the system provides two functions for its users:

1. The retrieval of information about their surroundings, and

2. The optional annotation of location for communal benefit.

It is hypothesized that LLI will enhance the urban experience, just as access to transportation dramatically altered the sensation and form of the city. By making inhabitants hyper-aware of their surroundings, they get the benefits of a small town citizen (omniscience of space and society) while possibly being situated in a much larger megalopolis with the social mobility and features that go with it.

Space Makes the Information Sea Legible, Distributed Data Remembered

With over 2 exabytes (2,000,000,000,000,000,000+ bytes) of information created annually [15], connecting this chaotic sea of information to physical space needed an organizing principle if it was to be as useful as data-rich. Inspiration for the construction of digitally overlayed information came from urban planner Kevin Lynch, who sets a metric by which we can recognize a good urban place: *one that is both legible and remembered* [39]. We use Lynch's aphorism as both a goal and evaluation criteria for this work. Similarly, LLI was built with the assumption that the challenge for the sustainability of digital information on a global instead of computational time scale will be to ensure *legibility* for its human navigators. Said another way, how can we constantly see what we need to see? To accomplish legibility and avoid losing ourselves in the information sea, rather than having access to all virtual information from everywhere in the city, *LLI utilizes space as an asset in narrowing the amount of data available at one time.*

The second revelation from Lynch came from expanding his concept of *rememberance* from something an individual does, to something that society does. This expansion was fueled by the rise of the collective powers of retention that the World Wide Web ("WWW") has engendered. With the WWW and corresponding search applications such as Google [28], it no longer matters that an individual has a particular piece of knowledge— what matters is that that knowledge is accessible, collectively archived. To acheive this style of rememberance, LLI adopts the WWW's promotion of distributed information storage and its encouragment of bottom-up data entry. Indeed, it is the distributed nature of LLI that is the novel element amongst other augmented reality system designs.

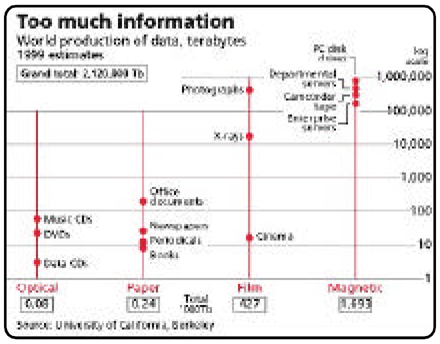


Figure 1-8. World production of information.

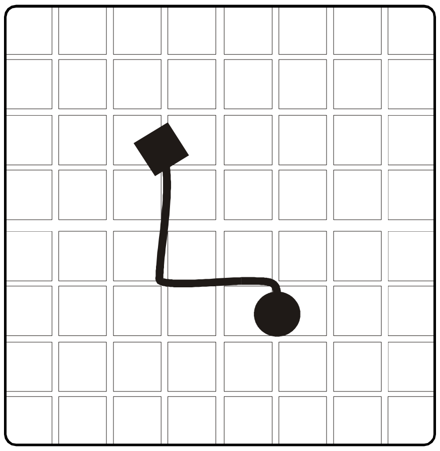
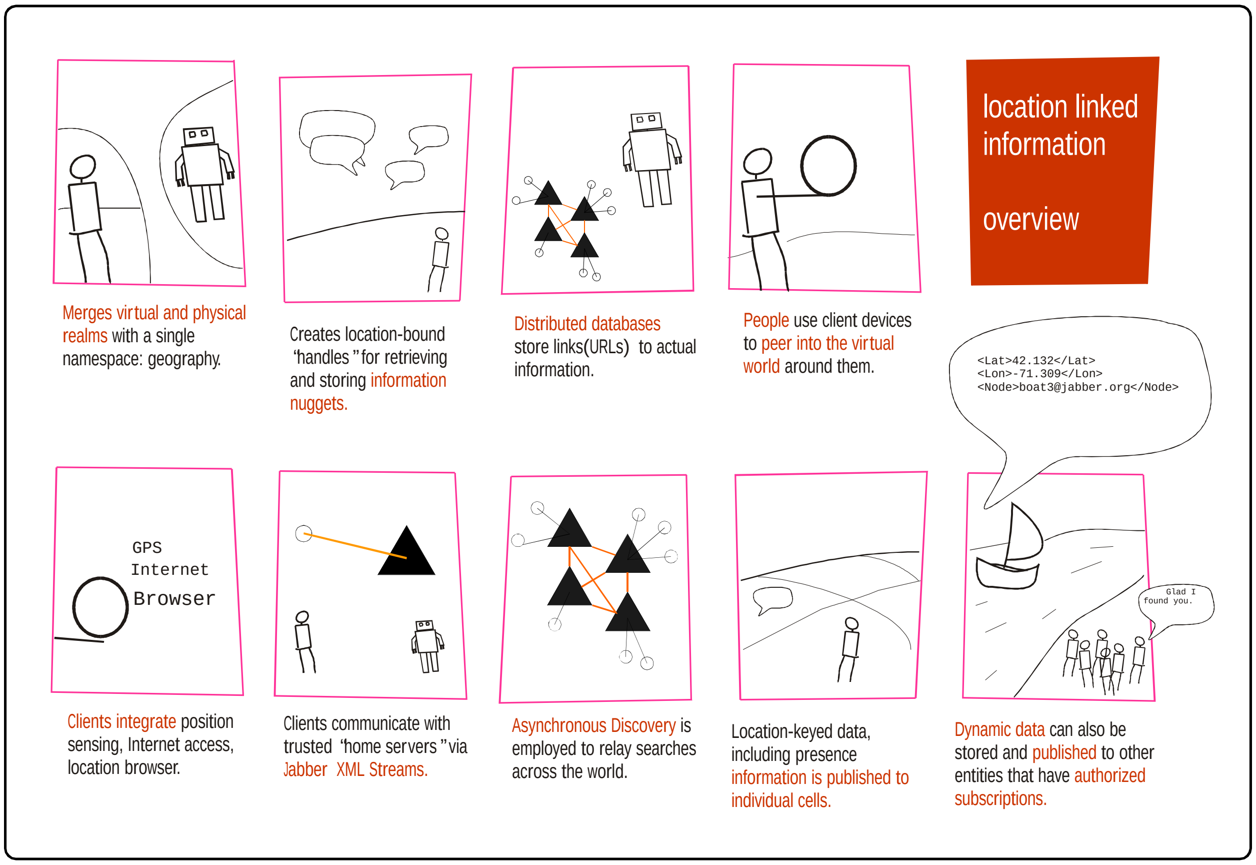


Figure 1-9. The city grid helps organize space and makes navigation easier.

Figure 1-10. City streets, the ultimate theatre of experience, can be difficult to navigate, especially if you are in unfamiliar territory. While this street in Taipei is chaotic, it is meaningful for its native inhabitants, however unweildy and confusing for a tourist attempting to find his hotel.

LLI Overview Cartoon

The LLI implementation will be detailed in chapter four, but a brief overview will set the stage for future discussions. Note from the overview cartoon below, the emphasis on distributed protocol and the social control of sensitive content via subscriptions.


location linked information overview
Merges virtual and physical realms with a single namespace: geography.
Creates location-bound 'handles" for retrieving and storing information nuggets.
Distributed databases store links(URLs) to actual information.
People use client devices to peer into the virtual world around them.
GPS
Internet
Browser
Clients integrate position sensing, Internet access, location browser.
Clients communicate with trusted 'home servers" via Jabber XML Streams.
Asynchronous Discovery is employed to relay searches across the world.
Location-keyed data, including presence information is published to individual cells.
Glad I found you.
Dynamic data can also be stored and published to other entities that have authorized subscriptions.

```
<Lat>42.132</Lat>
<Lon>-71.309</Lon>
<Node>boat3@jabber.org</Node>
```

Figure 1-11.

Thesis Roadmap

Like the city itself, software is an amalgam of numerous independent and complex parts whose exact inter-relationships are more experientially constructed than scientifically composed. The rest of this thesis is the narrative of the LLI software, starting with an example usage scenario in chapter two. This takes us to chapter three, which provides a technical context, looking at past, influential projects. Chapter four examines the construction of the LLI system, including both design guidelines and specific implementation details. Chapter five concludes by taking the trajectory of computation in the city further, looking at the impact of the introduction of sophisticated digital organisms. As a contextual aid, the appendices include some projects on collaborative mapping and data analysis that were undertaken during my exploration of this space, but which are adjacent to the main narrative.

Figure 1-12. George Orwell's *1984* portrayed a time that technology and society were integrated with less than optimal results.

Figure 1-13. LLI aims to integrate the physical and virtual realms, creating a rich urban experience.

CHAPTER TWO: EXTENDED USAGE EXAMPLE

"A good example is the best sermon."
— Ben Franklin

What can you do in the Smart City that you cannot do at present? What are the compelling reasons someone would want to go to the trouble of actually carrying out the work outlined in this document? The LLI system outlined in chapter four is an infrastructure project, and so by its nature is generic. Are there specific examples of usage that LLI attempts to make possible? These are some of the questions this section will answer.

The Dark and Stormy Night: A Possible Usage Scenario

It is a dark and stormy night. You are walking home from a stressful day at the office. The weather is so horrible that you duck under an overhang, pull out your PDA, and tap on the "Janus Map Browser" button to see if there is anything you can glean about the vicinity from Janus' dynamically generated map of the city. You see a plan of your current location with nearby streets surrounded by a flood of tiny multicolored dots of various levels of translucency that make the map look somewhat like a Seurat painting. As you move your pointer over the dots, they identify the type of information they represent— with iconic classifications that vary from location identification to historical text for tourists to alerts and public warnings. A click of a dot follows the link to more information.

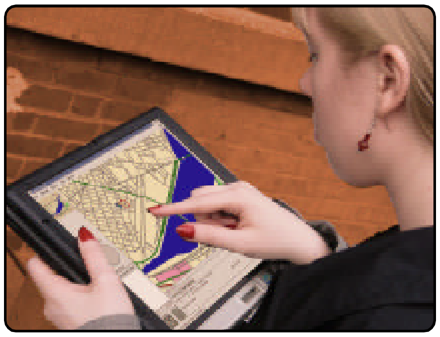


Figure 2-1. Using the Janus application to better understand the current activities available in an area.

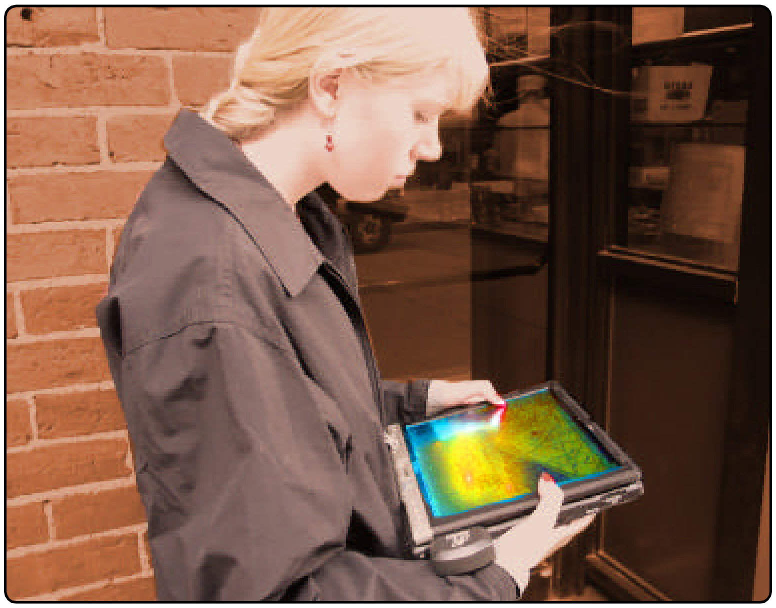


Figure 2-2. Viewing with Janus, a dynamic map application that serves as the gateway to LLI.

Upon release, you are shown only locations with activities— all the other marks and comments disappear. Most of the information includes things you have done before, but one bright circle a few blocks away labeled "Open Yoga Tonight" interests you enough to click on it. Your PDA then shows you the class schedule from the studio's Web site, and after reading other people's annotation nuggets in the area around the entrance to the yoga center, you decide to take this opportunity to explore this unknown part of your neighborhood. After clicking on one of the annotations you realize that it is not really a yoga participant's comment as it says it is, but Location Spam— an advertisement for the coffee shop across the street. With two taps of your PDA, you quickly correct the classification and demote its relevance so others in your position will not have to look at it.

Before leaving the dry safety of the overhang, you check the screen to see if any of your friends are nearby, which sadly they are not. You configure a beacon agent to watch for any of your friends who walk by the overhang in the next twenty minutes, automatically sending them the message:

```
"Hey.  I decided to go to the Yoga place, but
should be done at 9. If you want to meet at the
oxygen bar, I'll be there 'til 11."
```

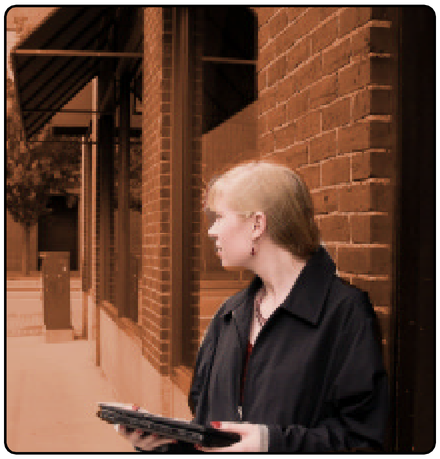


Figure 2-3. Utilizing a beacon agent to send a message to any friends who walk by in the next hour.

Figure 2-4. LLI allows you to see further, exploring your surroundings by doing things such as viewing available apartments in the area.

On your way through the unexplored streets, your Janus satisfies your curiosity by showing you all the apartments that are available for rent on the street, but you decide that the area is too expensive.

Directly before arriving at the street of the Yoga Center you begin to feel uneasy and change the Janus view to show a video feed from a Web cam that a resident has pointed outside his window a few houses away, which reassures you that the area is safe from sinister characters and it is just the rain and your overactive nerves at work.

At the entrance to the Yoga Center you realize that you are ten minutes early and so bring out your PDA and enter the *near-chat room*, setting the loudness radius to be a 1/10 of a mile. There are a few discussions going on about various events of the day. You interject:

```
“Has anyone gone to the Yoga Center on Harvard
Street?  I’m about to go, is it worth my money?”
```

To which a reply comes in a few seconds:

```
“Yeah, I go on every Tuesday.  Don’t worry about
it and have fun.”
```

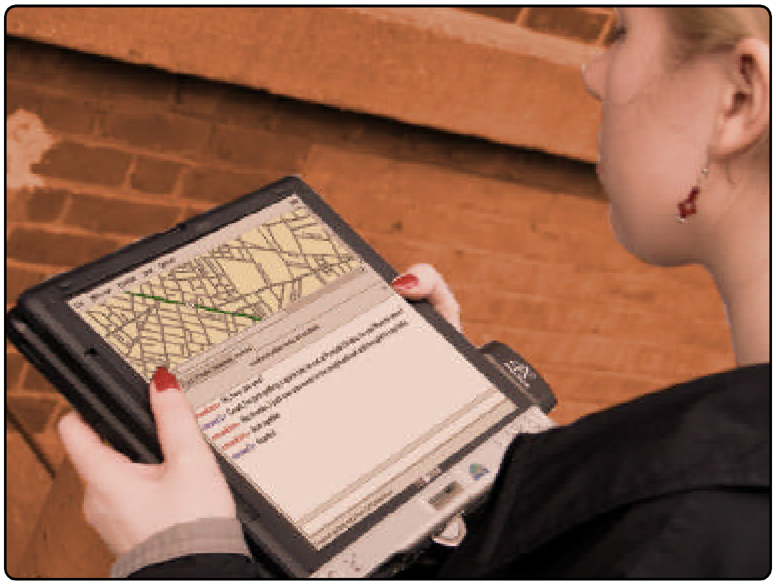


Figure 2-5. The near-chat room allows you to communicate with people who are in a close proximity, but which might not be immediately visable.

The Dark and Stormy Night: Dissected

In the example above a few key events are worth expanding upon, as they embody general principles and will be revisited again. Among them are:

1. Separation of presence from content.

2. Unification of varied endpoints through standardized protocol and coordinate system.

3. Integration of information from a multitude of sources and networks.

4. Subscription-based access control of information streams.

5. Communal content creation and annotation.

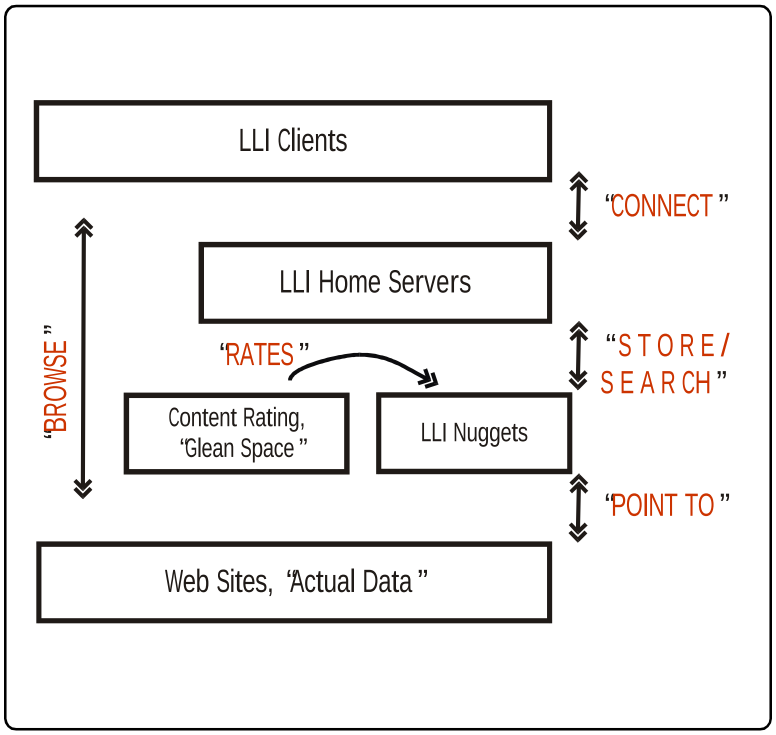


Figure 2-6. LLI component interaction overview. LLI clients connect to their home servers, which handle storage and search requests for them. LLI Home servers keep nugget data and content rating data. LLI clients fetch the "actual content" directly from the URL pointed to by the LLI nugget.

Separation of Presence from Content

In the above scenario many different devices and information streams have been codified with a standardized location and presence protocol, allowing these devices to know who was nearby and thus with whom they could "talk." This is a sort of "who's in my neighborhood" protocol where devices acknowledge each other as well as any humans and their agents, irrespective of the content of their data streams. For example, when feeling uneasy, the Web cam's data stream was tapped into, but before this could happen its presence was sensed and made available on the Janus' display. While the protocol that devices use is standardized so that they can intercommunicate, their functional content streams might vary widely. This division between presence and content is central to implementation, as we shall see in future sections.

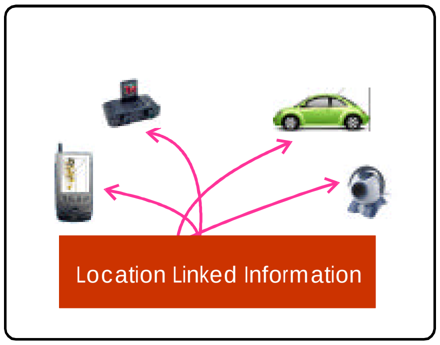


Figure 2-7. The LLI protocol integrates numerous data creation and access devices with a unified protocol.

Unification of Heterogeneous Devices through Standardized Communication

Besides the separation of content and presence, there exists a built-in allowance for a wide variation of "endpoint devices." In the telephone infrastructure, the endpoints started out as human-operated telephones, but later expanded to include modems, automated attendant systems, and voicemail boxes. Similarly, in the scenario above, there exists a diverse possibility for nodal devices. Some of these nodes might include graphical interface applications, such as the Janus, digital agent bots programmed to answer as a proxy for a human or to be activated by a condition being met (such as someone's presence), as well as sensor-type laden content providers, such as Web cams, in-road speed sensors, or facial recognition cameras. While these devices all vary in function, they are connected via a shared protocol which has built-in flexibility of content—just as you can say anything on the telephone, the protocol is designed to allow any type of data to flow through it.

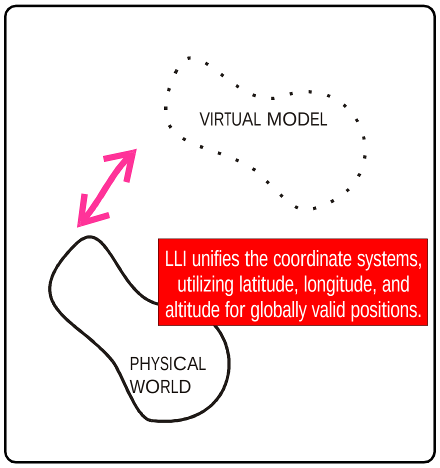

VIRTUAL MODEL
LLI unifies the coordinate systems, utilizing latitude, longitude, and altitude for globally valid positions.
PHYSICAL WORLD

Figure 2-8. LLI unifies coordinate systems, using latitude and longitude in both virtual and physical devices.

Integration of Networks and Physically Located Sources

Closely related to this concept of variation of function for connecting devices, are the permissible network typologies that can be employed. In the scenario above, it is imagined that the computer devices where this virtual information is stored and processed are owned by a wide slice of the public at large— from the Yoga Center advertising its location to a neighbor offering the view from his Web cam. Just as the Smart City is layered on existing cities, the technical implementation is built on the tradition provided by the Internet. Physical access to data is built on top of a global transport protocol, such as TCP/IP. Devices on this scenario's location network might connect via dedicated high-speed links such as DSL or Cable modem, or via shared typologies such as Wi-Fi. Consequently, because the device's communication backbone is the Internet, there is no enforced correlation between the physical storage or route of data streams and global location— a tourist information database in New York could just as easily provide historical information about your neighborhood as one physically located there. Because of this ambiguity of source, *trust and social moderation are required ingredients for usability and social acceptance*.

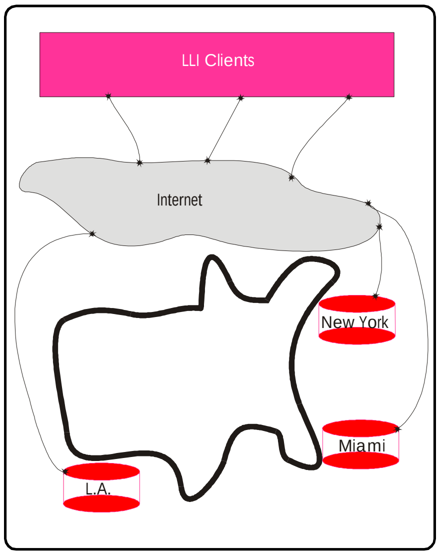


Figure 2-9. LLI clients are connected with geographically disjoint servers and network typologies through the Internet.

Subscriptions and Access Control

For example, when requesting that your friends be notified of your location for the next few hours you are granting this grouping of friends access to your position data. Implicit in this action is a trust that the digital infrastructure will actually obey your wishes and propagate your location data only to those whom you have recognized. The implementation outlined later will build on this concept of a dynamically constructed trusted relationship (as is implied by "subscription") so that data access and control is both rigorously enforced and fluidly granted.

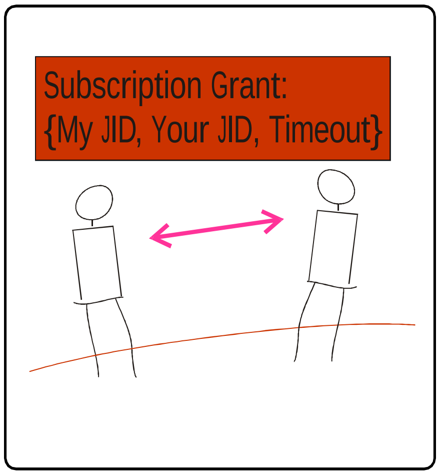


Figure 2-10. Subscriptions are granted by telling your home Jabber server your unique id, along with the id of an entity to receive the data and an optional timeout value.

In the stormy night scenario, you re-contextualized a piece of information as being an advertising message rather than a review of the building in front of you. Besides this democratizing ability to shape the informational environment presented, it should be pointed out that this virtual landscape is culled from possibly thousands of different sources, making it more difficult for one faction to completely control the virtual space. Of course, one of the motivating reasons for creating this kind of decentralized system is to allow anyone to enter information into the network, thus making it richer.

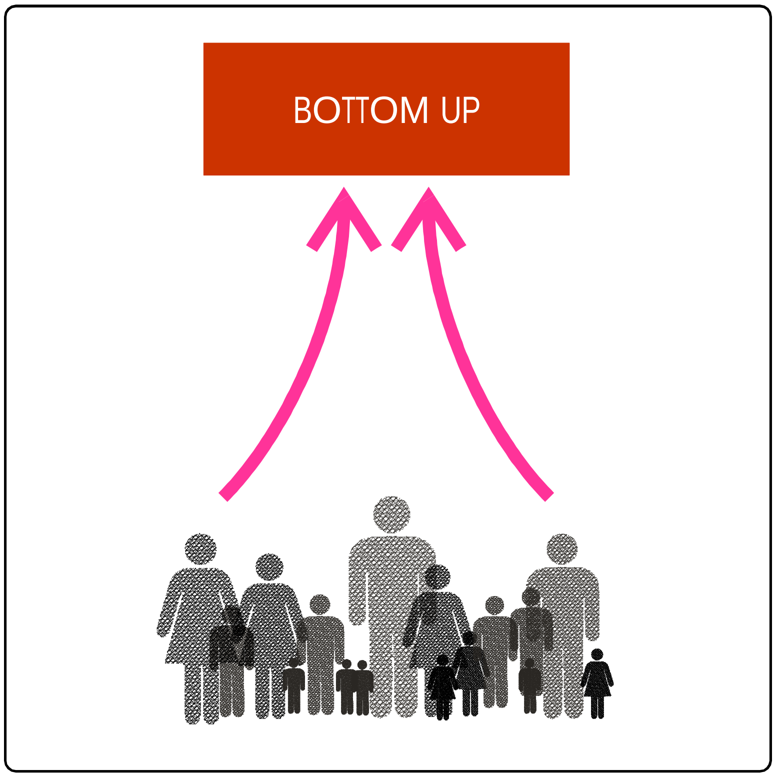


Figure 2-11. Data entry and storage is 'bottom-up" in LLI, meaning that it is assembled from many different sources rather than being kept in a central location or entered by one entity.

CHAPTER THREE: DESIGN CONSIDERATIONS & INFLUENCES

"It takes a thousand men to invent a telegraph, or a steam engine, or a phonograph, or a photograph, or a telephone, or any other Important thing— and the last man gets the credit and we forget the others. He added his little mite— that is all he did."
— Mark Twain

Figure 3-1. Geocachers find hidden treasures inside boxes such as this by utilizing a GPS to navigate to a latitude and longitude.

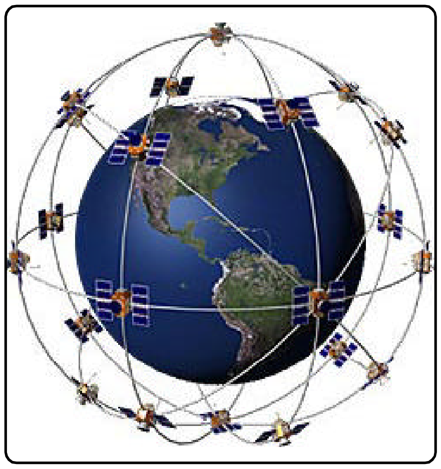


Figure 3-2. The Global Positioning System is made possible by a group of 24 satellites that orbit the Earth, transmitting a time signal. GPS receivers calculate the time difference between satellites to determine their latitude, longitude, and altitude.

The childhood game hide-and-seek received a high-tech makeover in May 2000 when the sport *Geocaching* was first played [24]. Instead of excited kids running through the house looking for their friends, grown-up geocachers visit the geocaching.com web site where lists of objects and their locations can be found. With coordinates in hand, the hunt begins and the GPS-equipped geocacher navigates to the hidden treasure. When the treasure is found, they hide it in a new location or swap the treasure and update the web site.

Geocaching served as an impetus that got me thinking about location and the seemingly limitless possibility of "the real world." Having spent much of my time indoors and at a computer, the concept of exploring the real world *and* having access to the Internet was compelling enough to begin tinkering with the technology to make it happen. The communications hardware was readily available, with a wide array of GPS receivers to choose from. Luckily I had just installed a Wi-Fi access point on my roof, and was able to wander around my neighborhood and have Internet access. What I was surprised by, however, was the lack of software to integrate with the GPS. Certainly there was a large array of "you are here"-type map programs, and some good navigational aids, but there was very little in the way of general software libraries to program with. I thought that this was a simple enough request, for I just wanted a means to associate digital messages with public places so that I could do things like create my own geocache or annotate a place.

So I set out to write some software tools— which became LLI— and release them as open source software to fill the void for others with my tendency towards computational wanderlust. This chapter looks at some of the previous experiments in merging the physical and virtual realms which influenced my design. Next, I outline the possible reasons that location, as a trend, has yet to "catch on," utilizing insights from prior work. After this, I point out some of the precursor technologies and trends without which this study would not have been practical. Finally, I conclude the chapter by giving a general overview of the design constraints and key functions for LLI, which will be expanded in more technical detail in chapter four.

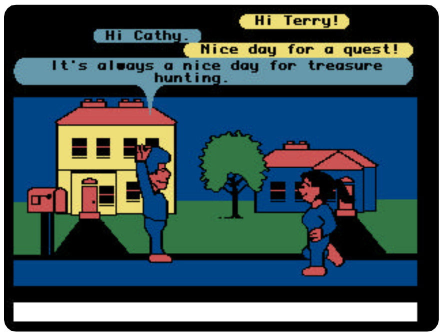


Figure 3-3. Lucasfilm and Quantum Online Services created a virtual, multi-user environment in 1985 for the Commodore 64.

Pioneering Work

Conceptually, linking information with physical space is not a novel idea— signposts, maps, epigraphs, murals, digital marquees, graffiti, cave paintings, and architecture [9] are all part of a class of asynchronous communication techniques that couple social cues with the environment [42]. With such a wide range of past devices and uses it is easier to untangle LLI's lineage by separating work into three categories, based on their major contribution: *interface*, *infrastructure*, and *collaboration*.

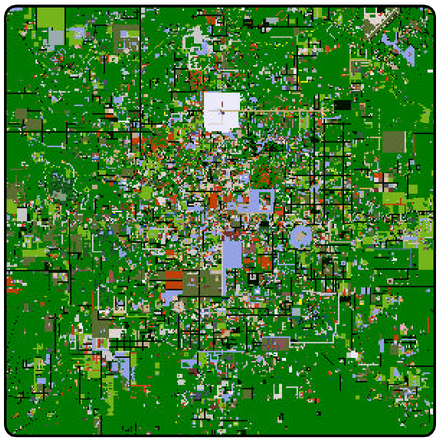


Figure 3-4. Plan of the 'AlphaWorld' cityscape, an early, totally virtual, user-constructed environment.

Interface

Interface projects attempt to answer the problems surrounding the interaction between the digital and physical domains. Ivan Sutherland's work in computer graphics in the 1960s proved to be the genesis for generations of computer interface programs [52]. Sutherland started the research into *augmented/virtual reality* with

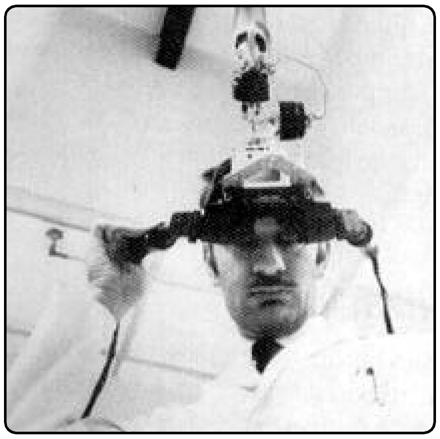


Figure 3-5. Ivan Sutherland's dream of a "kinesthetic display" fathered a generation of research in mixing the virtual and physical realms.

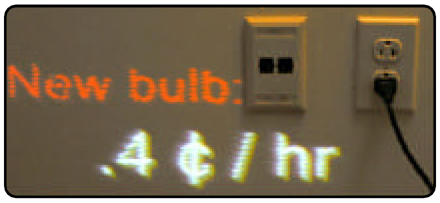


Figure 3-6. Devices like the IBM Everywhere Display can project images from computers onto the physical world, providing a lightweight solution to information display.

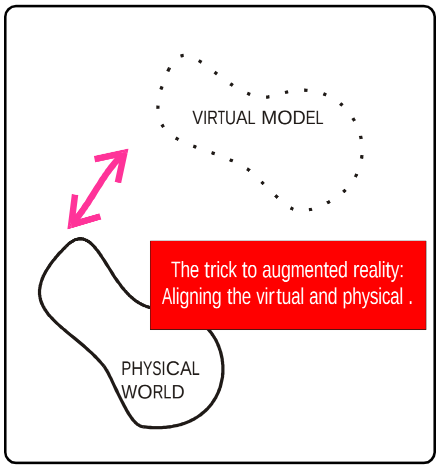

VIRTUAL MODEL
The trick to augmented reality: Aligning the virtual and physical .
PHYSICAL WORLD

Figure 3-7. The alignment problem of augmented reality.

his implementation of a "kinesthetic display," a head-mounted display unit that paints a three-dimensional display over the "real world". From Sutherland's work emerged a variety of projects which improved upon his technical foundation, adding higher resolution and color graphics, improved object occlusion, and decreased physical weight [6].

Decreases in the size of enabling technology migrated augmented reality systems from bulky computers into embedded, wearable and even fashionable [22] devices. These *wearables* still attempt to interface between the digital and physical worlds, but do so on a more personal, individual scale [5]. Alternative approaches to augmented reality include projecting into the environment with devices like the IBM Everywhere Display [55], decreasing the size of the per-person technology to zero.

What strikes me the most about much of the interface work is that its direction of orientation feels backwards. By this I mean that the experiments seem to start from the perspective of the computer and work towards the physical environment and the human. The approach is that of combining a model of the physical world together with a model of the virtual, and the 'trick' is to get these two models to align, whether that alignment is measured in terms of color depth, perception, or motion. It is clear from the decades of work in augmented reality interfaces that this alignment is a problem that I had little chance at solving— so I sidestepped the problem and settled the alignment problem by standardizing the coordinate systems between virtual and physical realms. Furthermore, I reduced the interface problem from placing a three-dimensional object in three-dimensional space, to one of maintaining one-dimensional points situated by three-dimensional coordinate properties (latitude, longitude, and altitude), thus drastically reducing interface complexity.

The second category of work includes devices that build an infrastructure for supporting location-based services and functions. These projects form a communications backbone from which location applications can tap into. Each successive generation of location infrastructure finds itself specifying smaller devices, capable of scaling to greater numbers of users, with less centralization.

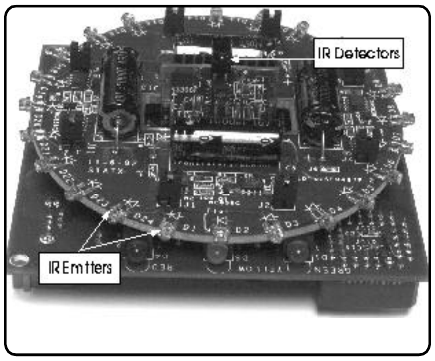


Figure 3-8. The PARCTab transceiver.

Active Badge by Olivetti and its sister project PARCTAB by Xerox were first-generation location systems which relied on globally unique ids and a centralized storage architecture to manage the locations of individual devices. In the PARCTAB system, an office environment was divided up into cells, with one Tab transceiver located in each cell. Transceivers broadcast infrared pulses which end devices would use to combine with a model of the layout to know where they were. The infrastructure could then be queried to know the location of any given Tab. The first use for these systems was for in-company telephone receptionists to know the location of individuals so they could route calls accordingly.

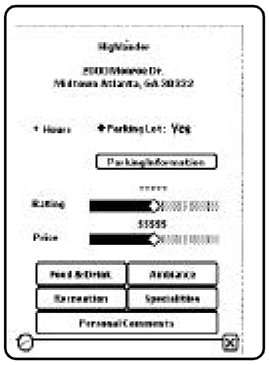


Figure 3-9. PDA Interface to Georgia Tech's Cyberguide.

Another application was Georgia Tech's *Cyberguide,* developed in 1997 as a digital tour guide. Cyberguide showed visitors around the researcher's laboratory, relaying pre-programmed tour messages to handheld computers that had knowledge of their current position. Cyberguide integrates the functionality of a cartographer, librarian, navigator and messenger to provide the experience of the tour [1]. Like other augmented-reality systems, the Cyberguide had a virtual model of the lab that it used to determine position and determine the correct message to display.

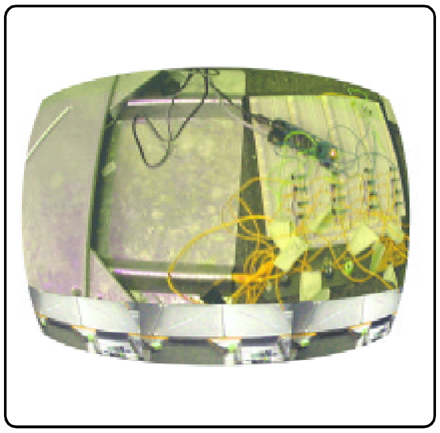


Figure 3-10. The Social Floor project at the MIT Media Lab embeds the electronics in the environment instead of requiring the user to be loaded with devices.

Subsequent lab tour applications include the Social Floor, built by the Context-Aware Computing Group at the MIT Media Lab. Just as Cyberguide utilized a model of the lab to determine what actions to take, the Social Floor relies on a pre-programmed knowledge of the layout of the laboratory to coordinate the projection of videos onto the lab's surfaces. What makes the Social Floor novel is its utilization

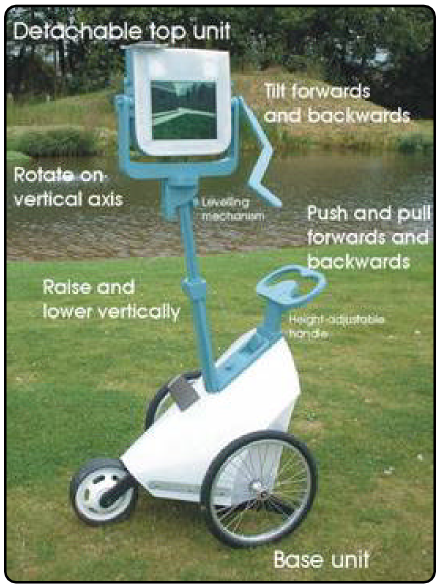


Figure 3-11. On the other end of the spectrum from the IBM Everywhere Display or the Social Floor is the Augurscope. The Augurscope provides a physical interface between the environment and a virtual model of the land as it existed in medieval times when Nottingham Castle stood on the same site.

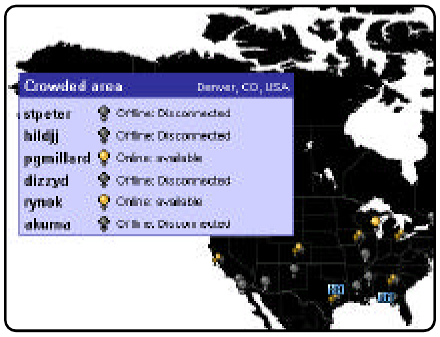

Crowded area
Denver, CO, USA
stpeter Offline: Disconnected
hildjj Offline: Disconnected
pgmillard Online: available
dizzyd Offline: Disconnected
rynok Online: available
akuma Offline: Disconnected

Figure 3-12. The Jabber World Map by Ralph Meijer builds on existing infrastructure, combining instant messaging presence with physical location in a web-based map representation.

of low-cost capacitive sensors embedded in the floor to determine a person's location, allowing a computerless visitor to stroll around the physical space in a natural way, getting a tour just as they would from a human guide. This interaction style is quite compelling, for the environment itself seems to be changing to meet your needs, and this experience of an active environment was greatly influential in the construction of LLI.

After the first location infrastructures mentioned above, later projects focused on limiting the dependence on a model for mapping between real and virtual realms. For example, Hewlett Packard's Cooltown introduced "nomadicy" by promoting the discovery of devices rather than reliance on a model of where events occurred [13]. In Cooltown, beacons tagged to devices would broadcast URLs that local devices could use to create a model of what is in their area. A human lost in the city might ask others "Where am I?", however in Cooltown it is the city that is constantly saying "You are near this device, accessible at this URL." The alternative approach, where the technology queries the environment to ask where it is, is taken in the more recent Web Feature Services ("WFS") standardization effort, which utilizes HTTP transported XML (actually a subset branded as Geography Markup Language, GML [25]) to tie together geographic "coverages" to features (maps, gis layers, etc.) [59]. A similar influence in the distribution of data and discovery of environmental abilities is MIT's Project Oxygen [47], which itself covers a wide range of technologies, generally stepping toward a world where humans interact directly with their environment.

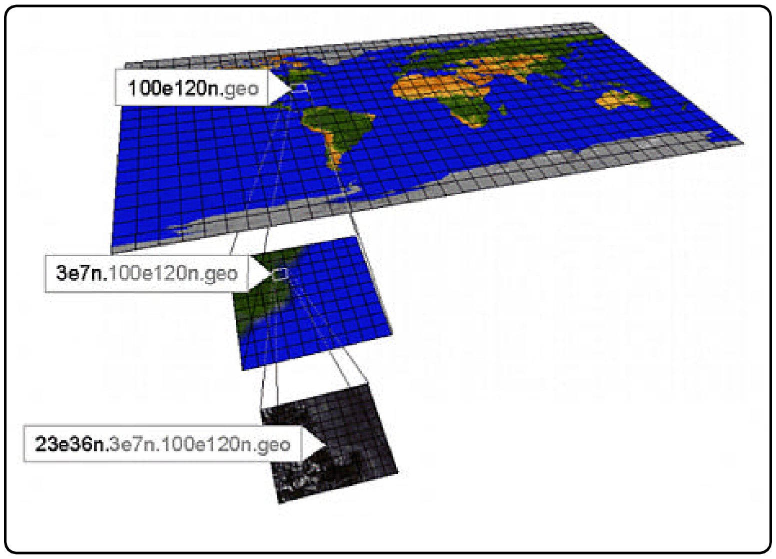


Figure 3-13. The GeoWeb project segmented the earth into addressable cells which were then given DNS names which could be delegated to servers on the Internet.

Another type of project altogether, and an important influence for LLI, the GeoWeb project attempted to utilize the existing infrastructure of the Internet for associating data with geographic location. To do this, GeoWeb divided the world up into cells (Figure 3-13) and maintained a top-level Internet domain of .geo which could then associate an IP address with a particular geographic cell. For example, "20e30n.geo is a hostname which identifies the 10-degree x 10-degree cell whose southwest corner is located at 20 degrees east, 30 degrees north" [14]. Specialized geo clients would then access the geo server via the web, retrieving an RDF meta-data file from the URL http://20e30n.geo/. ICANN, which controls top-level domain name creation, denied the .geo addition, so the project was not able to be realized. However the cell-structure, distributed data storage, and meta-data concepts were either directly incorporated into LLI, or greatly influenced by the GeoWeb.

Figure 3-14. GeoWeb imagined both hand-held application interfaces, such as above, as well as map-based information displays.

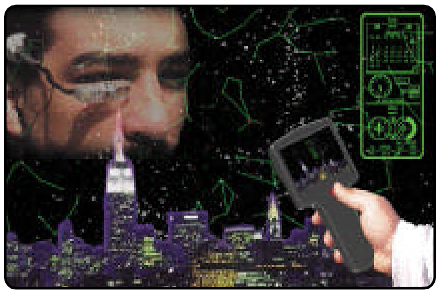
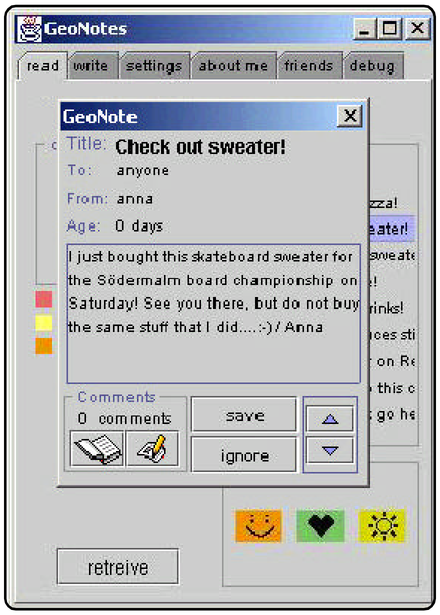


Figure 3-15. An example usage scenerio in Worldboard which integrated devices, space, and virtual models.

The final entrant into the infrastructure section is WorldBoard, “a proposed global infrastructure to associate information with places and ultimately to provide people with enhanced information perception services.” [62] WorldBoard imagined itself as a sort of planetary chalkboard where people could leave messages and associate them with any place on the planet. Of the many contributions of WorldBoard is its dual design goals of a) being operational from a planetary scale from the start, and b) being so simple that people actually use it. The scale and simplicity of WorldBoard, as well as the clear vision for how technology can impact lives of ordinary people are noteworthy, as was Spohrer’s analysis of the social factors necessary for his concept to be adopted:

> “Will this idea catch on? Or will putting information in places merely be an oddity, a technological ‘side show,’ that never quite worked right or had enough utility to become a truly viable global information service? Perhaps negative social implications will be discovered that limit adoption...”



Title: Check out sweater!
To: anyone
From: anna
Age: 0 days

I just bought this skateboard sweater for the Södermalm board championship on Saturday! See you there, but do not buy the same stuff that I did....:-) / Anna

Comments
0 comments

Figure 3-16. GeoNotes interface allowed users to leave notes for each other in either public or private modes. These type-less messages are a little conversational and could quickly accelerate toward cacophony. Because of this experience, nuggets in LLI are *typed*.

It is this social thought that motivates much of the philosophical discussion of chapter five, for as he points out, society must be a part of the discussion for technologies to be absorbed and welcomed.

Collaboration

The final category of influences focuses on asynchronous communication keyed to location to promote collaboration. This group of projects is perhaps the most closely related to LLI, in that they have a similar goal of relating messages to physical locations, utilizing both of the infrastructure and interface work to create a human-centric communications experience.

GeoNotes is a Java application meant to be used in conjunction with a GPS device which provides latitude and longitude to a portable computer or PDA. A GeoNotes client contains a user interface for people to leave notes, storing these annotations for other GeoNotes users. Furthermore, GeoNotes has an interface for content rating and the sending of private messages. As the GeoNotes designers noted, "the success of such a system fundamentally depends on its ability to filter out irrelevant notes." [26]

GeoNotes utilizes the *Wherehoo* server and protocol for its data storage and retrieval. Wherehoo/Periscope, developed at the MIT Media Laboratory, were among the first location-infrastructure projects to differentiate between the data retrieval, storage and display functions by splitting the project into two parts, connected via a common protocol. Wherehoo, is a service for location storage and retrieval, and Periscope, is an example user interface and software agent. LLI borrows this modular design, with the LLI server/protocol being analogous to the Wherehoo server/protocol, and Janus to the Periscope interface.

Another collaborative project is E-graffiti, by Cornell's Human Computer Interaction Group. E-graffiti, like GeoNotes allows users to enter location-tied notes using mobile computing devices. E-graffiti uses the Wi-Fi network to determine position, associating the MAC address of an access point to a particular location. Users are then able to post both public and private notes to that access point so that others can later see the note. Influential in understanding possible uses for location is the result of a user study conducted for the project, which found:

> "The intended purpose of E-graffiti was to allow users to, in essence, communicate through locations, to use a location as a sort of proxy for information sharing. The appropriate model of use for the system would have been an annotation system where users could communicate asynchronously by annotating a location with relevant information. *However, users saw it differently. They used E-graffiti as a type of networked instant messaging or e-mail system. In fact, synchronous conversations between friends using E-graffiti was a tactic many students used to 'whisper' in class.*" [16]

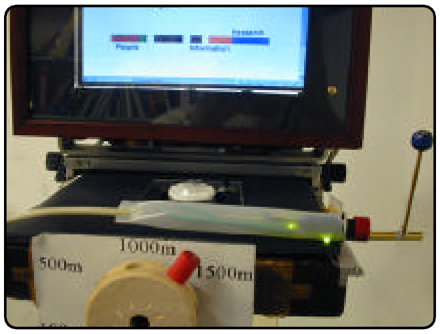
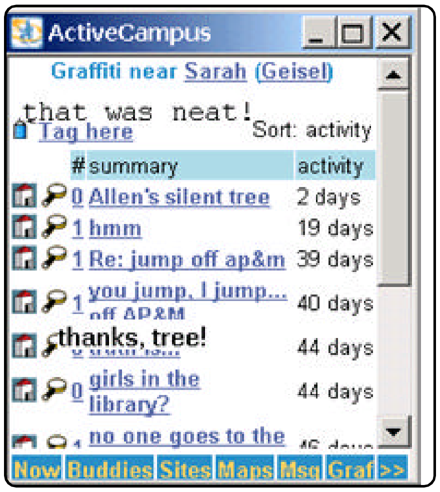


Figure 3-17. Periscope was a browser for the physical space, allowing you to navigate to web sites that were located near you by swiveling the display and adjusting a distance knob.



ActiveCampus

Graffiti near Sarah (Geisel)

that was neat!

Tag here Sort: activity

#	summary	activity
0	Allen's silent tree	2 days
1	hmm	19 days
1	Re: jump off ap&m	39 days
1	you jump, I jump... off AP&M	40 days
	thanks, tree!	44 days
0	girls in the library?	44 days
1	no one goes to the	46 days

Figure 3-18. UCSD's ActiveCampus (2003) incorporated messaging into the base functionality of the location system.

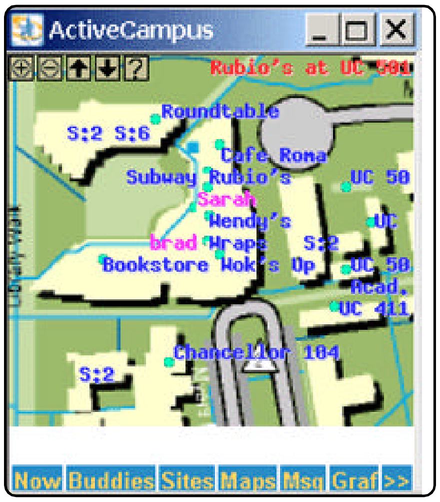


Figure 3-19. ActiveCampus displays at UCSD take the campus-based E-graffiti concept forward, incorporating maps and utilizing Jabber for messaging.

Location Location Location: Does Anyone Care?

Figure 3-20. The *34 North 118 West* project couples Tablet PCs with a tour of an area in Los Angeles, interjecting audio as users navigate the area.

Some evidence exists that people do care about the combination of location and technology, as recently a term has emerged to describe this genre: Location-Based Services ("LBS") are applications closely related to both mobile computing and augmented reality, but which explore the peculiar properties afforded by physical location [17]. Of course, LBS are a nascent area of study that is devoid of both a killer app and a sizable user base [53]. I would suggest that the reason for this lack of adoption is twofold:

1. Content development is costly and time-consuming [11]; and

2. Enabling technology for LBS is largely centralized and unable to organically grow.

Futurists touted LBS as an area ripe for explosive growth in the late 1990s [56]. To date, the technology has been marginalized, in use in several niche application domains (in-car navigation, tracking truck fleets, virtual tour guides), but despite its general potential it remains largely out of mainstream usage. Certainly ease of use and access is part of the problem, as Mark Weiser noted:

> "Desktop publishing... is fundamentally not different from computer typesetting, which dates back to the mid 1960s, at least. But ease of use makes an enormous difference." [60]

Figure 3-21. WordStar, an early PC-based word processor required control key combinations to move around the screen.

Anthony Townsend suggests that a possible reason for the lack of adoption is because only technologists have been involved in the deployment of these services. He suggests that those in the location professions— architects, urban planners, and geographers— need to be more closely tied to the construction process [54]. I take the view that the doors to real development of LBS have yet to be pried from the hands of technologists— the simple foundation for LBS targeted at the populace does not exist: current methods are too technologically heavy and do not empower the populace to take part. I do not mean to suggest that LLI itself is significantly less technologically mired— for it clearly is seeped in the technological tradition of MIT and the Media Lab— only to offer the decentralized/protocol driven/bottom up cues as a catalyst for reaching the no-tech tech nirvana.

Closely Related Enabling Trends

LLI is closer to public consumption and realization because of two innovations:

1. The maturation of mobile technologies;

2. The indoctrination of Internet rituals/the rise of the active consumer [20].

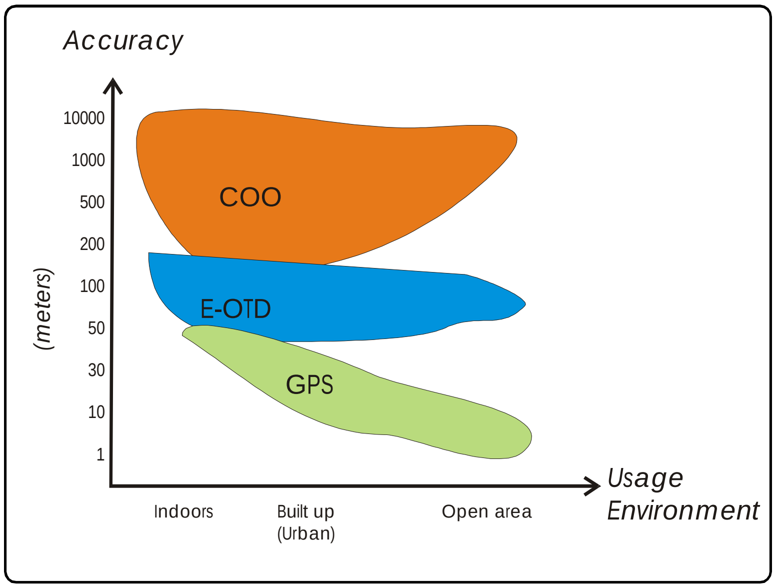


Figure 3-22. Accuracy of various positioning technologies in relation to the type of environment.

First, the deployment of location-based information applications (and by this I specifically mean outdoor location-based information, although the basic principles apply indoors as well) has recently been given three new classes of tools that promise to take its work to a new level:

Figure 3-23. The miniaturization of technology has made it possible for us to consider sharing our environment with devices like this *Garmin ique 3600* GPS/Map.

1. Precise electronic positioning systems (GPS, E911, Self Positioning Algorithms [29]);

2. Miniaturization of computation, communication, and display equipment; and

3. Ubiquitous, low-cost Internet network connections (GPRS/CDPD/Wi-Fi).

With these technological advances it becomes feasible for developers and end users to access and generate information that resides in both the city of bits, and the city of atoms [43].

Second, our society has begun to mirror itself after the prevailing media technology of the day: the Internet. Rather than being a television-fed, passive consumer society, we have become an Internet-guided, active consumer/designer society [20]. Having been given the opportunity to micro-define their consumption and social patterns, city dwellers have taken on the dual role of designer/consumer as they utilize technologies of mass customization to engage in personally meaningful activities [20].

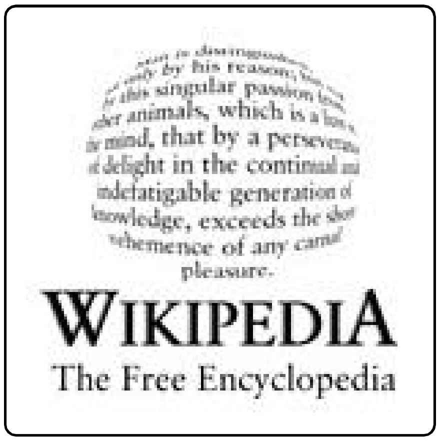


Figure 3-24. The wikipedia introduced communal content control and the benefits and problems that come with it.

Furthermore, the rise of the Internet has introduced social rituals for the manufacture and maintenance of asynchronous communities [42]. Netizens are accustomed to posting comments (the authoring of content) to forums like Slashdot [51] or the Wikipedia [61] where their messages are subject to the whims of the community's readers (rating and editorial discretion lies in the hands of the populace).

Another praxis, legitimized by the invention of the blog, is the acceptance of everyone as content developer— a concept at the core of LLI. Without these established social principles and trends it would be a monumental task to suggest such a cocktail of behavior changes.

Location Linked Information System Design

Functional Overview

Because of the technical nature of the implementation chapter, I wanted to give a slightly less technical overview here, so that those uninterested in the details of implementation can skip that section and not miss out on the general goals that drove the implementation.

As mentioned, the LLI system dynamically links a physical time/space moment with a *distributed* database containing "nuggets" of information related to that moment's surroundings. Embodied by an interactive, dynamic map, called the Janus, which serves as a bi-directional portal to cyberspace from space, the system provides two main functions for its users:

1. The retrieval of information about their physical surroundings, and

2. The annotation of location for communal benefit.

Design Techniques

However, the design as put forth is distinctive through its synthesis of the following notions:

1. Data entry is "bottom-up." There is no central server infrastructure that has to approve additions to the system. Anyone can setup their own LLI storage system and connect clients to it.

2. The digital agents in LLI trust the humans to get information "right" more often than "wrong," relying on humans for semantic and context-sensitive filtering.

3. The tuple (Time, Space, URI, Location type, Globally Unique ID) is the "primary key" of all data.

4. A hybrid virtual/real space is created, ("glean space") owned and managed by the public, for the benefit of the populace. Conceptually, everyone "owns" these location-keyed pointers.

5. All content, apart from that which is part of the glean space, is owned by private individuals and maintained on their servers.

6. Adding glean space content ratings is open and without restriction.

7. The concept of presence is integrated in the design and tightly managed by time-based access control subscriptions.

8. Implied social contract: sensitive dynamic data, such as the current locations for humans, is controlled by an explicit subscription system that implies a social contract between agents and humans.

What Does LLI Do?

Now that the general motivations for LLI have been covered we can focus on what LLI actually does. This list of specifications provides a sketch of the parts of the system.

1. LLI uses geography, measured in degrees latitude and longitude as the primary key linking the physical and virtual realms.

2. LLI is similar to augmented-reality systems, which overlay digital information on top of the physical world. Whereas augmented-reality systems typically concentrate on solving the user interface problem, *LLI attempts to solve the data access and search infrastructure issues*.

3. In LLI, users navigate the physical world with a variety of XML-speaking devices, discovering and leaving "handles" to information nuggets.

4. A distributed network of databases manages the information nugget pointers that are URIs to actual information.

5. Information nuggets themselves are position/time/type/URI tuples that lead the viewer to further sources of data.

6. People use client devices, like the prototyped "Janus" client, to peer into the virtual world around them. Client devices can come in many different form factors and be specialized for finding particular types of information. LLI clients will typically integrate position sensing (currently with GPS), Internet access (Wi-Fi/GPRS/CDPD), and a browser/user interface.

7. Clients communicate with trusted "home servers" via Jabber-encoded XML streams. Relaying requests through a home server (such as is done currently with email) could provide users with a more anonymous location-browsing environment.

8. LLI clients search for information via the Jabber asynchronous discovery protocol, created during this research, which relays search requests to other servers across the Internet.

9. In LLI, the world has been divided up into latitude/longitude-based cells. Location-keyed data nuggets are then "published" to individual cells.

10. Applications that can take advantage of this system include both those that wish to permanently tag an area (static nuggets), as well as dynamic object presences (dynamic nuggets). Dynamic systems could be used for vehicle tracking (air, car, boat, etc.), friend tracking, or anything else that varies with time. Rather than coupling dynamic nugget data to a fixed cell, it becomes associated with the entity that is in motion, and anything that has a subscription to that entity receives thedata .

CHAPTER FOUR: TECHNICAL IMPLEMENTATION

"If the work of the city is the remaking or translating of man into a more suitable form than his nomadic ancestors achieved, then might not our current translation of our entire lives into the spiritual form of information seem to make of the entire globe, and of the human family, a single consciousness?"
— *Marshall McLuhan*, Understanding Media: The Extensions of Man

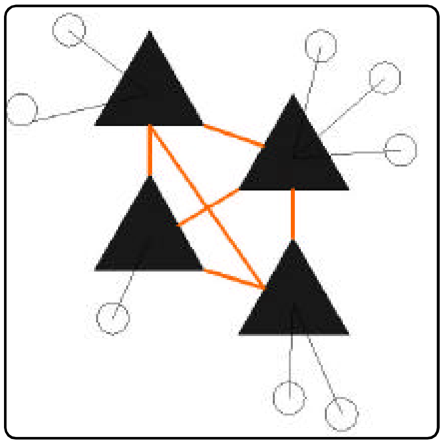
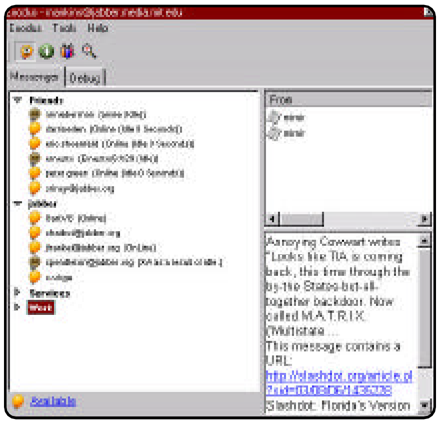


Figure 4-1. LLI architecture overview. Triangles represent LLI servers, circles are clients.



Annoying Cowart writes "Looks like TIA is coming back, this time through the by-the States-but-all-together backdoor. Now called M.A.T.R.I.X. (Multistate ...
This message contains a URL:
http://slashdot.org/article.pl?sid=03/08/06/1436238
Slashdot: Florida's Version

Figure 4-2. Jabber, originally started as an open, XML based Instant Messaging solution has grown into much more. Pictured is the Jabber client Exodus, used for instant messaging.

LLI: Architectural Overview

An overview of the LLI system shows two main parts, the server and client sides, connected via a standardized protocol. LLI servers are responsible for data storage, retrieval, and distribution as well as client message routing, subscription management, and authentication. Clients display data nuggets, interface with their users to relay their subscription requests, and receive location-based annotations. Clients might also be second-order clients, which integrate the LLI protocol into other application domains. Clients and servers communicate with a common protocol based on Jabber XML streams. Therefore, many different implementations of the LLI system could be built, so long as the XML spoken by the client and the server remains consistent to the scheme.

While the two main parts of the system are client and server, LLI is not strictly a client-server topology, but more of a hybrid between client-server and server-to-server or hub-and-spoke topology. Multiple clients connect to a single "home" server, which manages authentication and nugget storage for its clients. The LLI server is then connected to other LLI servers to form a virtual peer group. LLI servers in the peer group relay search requests to each other, with the home server responsible for relaying any search "answers" to their clients. Servers employ "store and forward" techniques to ensure message delivery.

Jabber Jabber Jabber

LLI's XML-based protocol is constructed on top of Jabber, but what is Jabber? The Jabber Software Foundation (JSF) defines Jabber as "an open XML protocol for the real-time exchange of messages and presence between any two points on the Internet." [33] Jabber was originally an open-source project, conceived by Jeremie Miller, to facilitate interaction between instant messaging systems, giving then closed-protocol instant messaging clients (AIM, ICQ, MSN, etc.) a common language to exchange messages. In time it became apparent that Jabber was more generic and useful than just as a protocol for instant messaging. As a community formed around Jabber, a group lead by the JSF brought the protocol to the Internet Engineering Task Force (IETF) for consideration as an IETF-sanctioned messaging protocol. The IETF Working Group is in its final stages of standardizing the core Jabber protocol, which it has rebranded as "XMPP," the eXtensible Messaging and Presence Protocol [19].

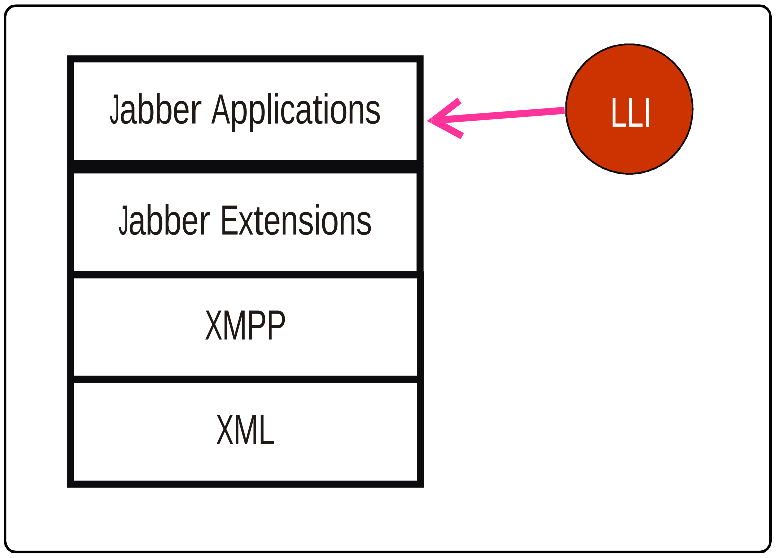


Figure 4-3. LLI data stream is a Jabber application, conceptually layered on top of Jabber Extensions, XMPP, and XML.

Jabber, then, is the term for the collective grouping of applications that make use of its XML streams to communicate, whereas XMPP is the standardized, base protocol which all Jabber applications employ. All Jabber applications share XMPP as their least common denominator. The Jabber community has made a number of libraries for different client hardware and software architectures, making it relatively easy for application developers to put their hands into their digital toolboxes and come up with a way to make two things talk to each other without doing a lot of worrying about security, message routing, robustness, or future extensibility.

Why Jabber?

Certainly I could have designed much of the functionality required for LLI from scratch, possibly making it more domain-specific and custom-tailored to the task at hand. The reason I did not take this approach (beside an obvious time constraint) is because this would be reverting backward to a state where all the advantages I am about to list would not be tested and robust. Said another way, we could all be programming in assembly language, but for a sea of reasons we do not.

Instant messaging has a tradition of trying to replicate physical-world structures: space is organized into chat rooms, similarly social cues such as an office door being open or shut is reproduced virtually with the online/offline states. As might be imagined, this tradition is beneficial in creating physical-world structures that utilize virtual data. The instant messaging tradition has seeped into Jabber, giving it a functional head-start for our application. Jabber itself has evolved further than instant messaging, generalizing its protocol (to subvert protocol pollution) and allowing room for expansion. There are five key, explicit benefits for using Jabber in location-based applications that are worth exploring further: presence, synchronous asynchronicity, built in publish-subscribe, extensibility, and believable identity.

1. *Presence*. Presence, "the immediate proximity in time or space [46]," is a key component of defining physical space, especially those spaces that have dynamic elements, such as people. In physical space, people read or sense your presence by viewing you within a space— *we view presence from the perspective of the observer*. Who is present in the conference room? Why Stacie, Nick, Yohan, and Parul are. We can think of presence as a probed and intrinsic property of objects.

Contrast presence then with location, which describes an entity to a possibly spatially or temporally disconnected observer. Where an entity (like a person) is located is told from the perspective of that entity. Location can be described using a coordinate system (such as "I am at latitude 42.352N, longitude 71.088W") or by using semantic methods ("I am on Ames Street in Cambridge").

As central as presence is to the physical world, it is largely absent from the core protocols of the Internet. Because of the ubiquitous and ungrouped nature of the Internet, it might not be immediately clear what presence would mean in such a spatially devoid environment. Nevertheless, application-layer additions have created a de facto definition for presence on the Internet: *being accessible is presence on the Internet*. For example, the Netcraft Web survey probes the Internet in search of Web servers, which it tracks in both numerical terms (42 million as of July 2003. and uptime (wwwprod1.telia.com has the current record, having been "online" for 1,761 days) [44]. Similarly, the "Big Brother" utility for system administrators codifies the presence of servers, saying they are either online (green), in need of attention (yellow), offline (red), or unknown (purple) [7].

Jabber incorporates the concept of presence into its core protocol, as one of three top-level server-to-client XML tags (<presence/>, <message/>, <iq/>) [2]. A result of this low-level coupling of presence within Jabber is that it becomes feasible to construct "views" of entities in virtual space, for example, a view of people's presence on your roster, in a chat room, or "nearby" [57]. LLI extends presence packets to include location information.

2. *Synchronous Asynchronicity*. As Jabber's goal is to deliver messages across the Internet, it could choose to do this delivery synchronously or asynchronously. A synchronous connection would be similar to File Transfer Protocol (FTP), where an endpoint connects, and stays connected, to an FTP server during the life of the message transmission. Synchronous connections are expensive because they require constant resources, regardless of actual usage; but on the other side of the coin, synchronous connections typically have less overhead per message sent, and thus have slightly larger bandwidth potential. An example of an asynchronous system is the email (Simple Mail Transfer Protocol, SMTP) system, which receives messages to an email address (yohan@somedomain.net), stores it, and forwards it on to a destination's email server, when that server is available or not busy. Typically, asynchronous systems will be less expensive and more robust because they can process connections on their own schedule; of course, because of this property, their responses will not always be instantaneous.

Jabber is a hybrid, simultaneously supporting both synchronous and asynchronous methods, making it both robust and efficient on the upside as well as resource hungry and redundant on the downside. Entire papers could be written arguing the finer points of the above generalizations, however I bring the issue up simply to underscore the property of simultaneous synchronous and asynchronous messaging. An example should help clarify this oxymoronic statement.

A Jabber client, connected to her home Jabber server, will usually (it is not strictly required) maintain a constant connection, much like an umbilical cord from mother to child. The client authenticates with her Jabber Id, ("the jid,") and password, after which the XML stream is said to be authenticated and open. When the Jabber server receives messages destined for a jid (of top level tag matching <presence to="myjid@jabber.media.mit.edu"/>, <message to="myjid@jabber.media.mit.edu"/>, or <iq to="myjid@jabber.media.mit.edu"/>) it will route those messages to the client for further processing. Should the client go offline, the server will (optionally) store the message for retrieval when the client goes online again. Furthermore, because messages can originate from jids other than the server's (every entity in Jabber has a jid), the messages themselves could have been generated at the "present" or have been stored and forwarded from a "past" time period.

This ability to process messages without regard to time proves useful in LLI because it allows messages to propagate a group of servers who respond "when they get around to it." Similarly, messages can be left and retrieved by others without regard to time, just as signposts, buildings, and other asynchronous forms of communication do.

3. *Publish-Subscribe (Pubsub) Model*. Jabber has built-in support for publish-subscribe data distribution. Pubsub has its origins in content-based networking to allow asynchronous event notifications, such as implemented in the Siena project from the University of Colorado, or IBM's Gryphon system [3, 10]. These systems attempt to design a scalable data brokerage scheme, where "topics" (called "nodes" in Jabber) receive subscriptions by various interested entities. "Publishers" then publish data to the pubsub nodes that then gets relayed to all subscribers to that node. The distribution mechanism in a pubsub system is similar to that of an email mailing list, in that a mailing list contains a list of subscribers that get relayed data posted to the mailing list that may be restricted by an access control list. Besides this mailing list-like functionality, pubsub in Jabber contains a tree-based hierarchy, allowing jids to subscribe to an "inner" node of the tree rather than the root [40]. This subject-based filtering allows fine-grained data distribution. As an example, in the LLI system, nodes look like presence/yohan@jabber.media.mit.edu, generic/KLLYHUPYNTSZVSWKMNAXVT, or cell/40,-70:2,1:21,05:05,18 where the node delimiter is the '/' character.

LLI uses pubsub to allow interested parties to subscribe to a jid's geo-location and to receive automatic updates whenever that geo-location changes. Similarly, the pubsub code manages the node subscription process, giving fine-grained control over who can subscribe to your geo-location node, and for how long. This integration with a subscription system helps thwart Big Brother charges because access to the propagation of data is controlled by the data's owner.

4. *Extensibility*. Jabber is extensible through the addition of XML namespaces inside top-level (<message/>, <iq/>, <presence/>) tags. This allows us to "piggyback" on top of the Jabber architecture and protocol, reusing its time-tested servers and client libraries, extending the protocol to meet our requirements. Furthermore this expansion through namespace allows us to isolate our additions from others and still maintain legacy interoperability. Clients do as much as they can with the data given to them, ignoring any namespace they do not know what to do with. This concept is similar to parsers for markup languages such as HTML that are designed to ignore unfamiliar tags.

5. *Identity*. Built in, believable, identity functionality is central to the message routing and security of the Jabber system. All entities, be they human clients, artificial clients ("bots"), servers, or server components (an addressable add-on to the server, like the LLI component) have globally unique jids. yohan@jabber.media.mit.edu is a jid. Jids look like email addresses, in that they are generally created through:

```
"username" + "@" + "server hostname".
```

However jids have two additional, optional forms:

```
"username" + "@" + "server hostname" + "/" +
"resource",
```

or the server form:

```
"hostname".
```

In the full jid form, a resource is appended, making the jid look something like: yohan@jabber.media.mit.edu/Work. This form allows one entity to have multiple different instances, for example Yohan might also have the jid yohan@jabber.media.mit.edu/Home. Each instance has a corresponding priority, so that the server can figure out where to route messages to the simple jid form (yohan@jabber.media.mit.edu). Similarly, messages can be routed directly to a full jid, and only that resource will receive it (yohan@jabber.media.mit.edu/Janus), regardless of priority.

The hostname form of jids are reserved for Jabber servers and the pluggable components of servers. Each of these jids needs to be resolvable via the Domain Name System (DNS), as part of the login procedure for jids of this form involves a "callback" mechanism to thwart malicious uses like spamming before it starts.

All jids are authenticated before being able to receive messages, and thus there is some guarantee that when you receive a message it was actually from the purported source. Issues of identity rise to the surface as the jid could be used to tie a particular human entity to a virtual entity. Because of this coupling, and the general desire to maintain some degree of space-based anonymity it is important that we carry a high degree of confidence that who we think we are talking to is actually that entity.

The LLI Component

As mentioned, the LLI system is composed of two primary parts: the client and the server. The server is actually a Jabber server component. The Jabber server architecture is such that it allows numerous components to plug into the server, providing gateways to various applications or other messaging networks (such as ICQ, AIM, Yahoo, MSN, Zephyr, or SMTP) [34]. So henceforth when referring to the LLI server, it actually refers to the component of the Jabber server that handles the LLI requests.

What Do LLI Servers Do?

The LLI component in the Jabber server has the following responsibility:

1. *Answers search queries*. When clients or other LLI servers request information, attempt to provide an answer, propagating to other remote LLI servers if a timeout has not been reached.

2. *Maintains awareness of the LLI network*, making note of other LLI servers and storing them in a routing table for future searches.

3. *Stores location nuggets.* If a client requests, the LLI component will publish the nugget to the appropriate pubsub node.

4. *Relaying nuggets*. When new nuggets come in, relay to any jid subscribed to the node(s) the nugget is in.

5. *Become an expert on an area*, storing in the cache any external search requests that match the per-server defined "expert area" criteria.

6. *Manages payment* for restricted content.

7. *Cleans up*, reaping expired nodes and subscriptions.

8. *Manages nodes*, coordinating affiliations, creation, and subscriptions.

Answering Search Queries

When a search arrives at a LLI component, it first does access control verification, to make sure the search is valid on our server. If it is determined that access is allowed, the LLI component creates a generic pubsub node that it will use to "return" all the answers to the jid that requested the search. Searches are done asynchronously, and may return data immediately, or within some timeout period. This generic pubsub node takes the form generic/ PSEUDO_RANDOM_STRING_HERE, for example: generic/ KLLYHUPYNTSZVSWKMNAXVTYXOXKNUW.

The LLI component then subscribes the requesting JID to that generic node. This creates a channel between the LLI component and the requesting jid, allowing the answer messages to trickle in and get relayed as appropriate. Then the component performs a local search, determining if it has any data nuggets to publish to the node.

Next the LLI component walks its route table, determining if the search is within the route's "primary density zone." Each LLI server can have multiple bounding regions that specify certain densities of nuggets. At a minimum, each LLI server must have one bounding box that specifies the extents of its nuggets. Each region is specified as the corners of a rectangular bounding box, where the coordinates are expressed in latitude and longitude based on the WGS84 datum.

The search, if the expiration time has not been reached, relays the search to each remote LLI component route that is within the primary density zone. Each corresponding component will create a generic node, and return that node to our component, which connects the remote generic node to our local search node with a duplicate filter. As it is possible that remote servers will each contain similar data, the LLI component should filter out redundant data before sending it to the initial searching jid. Each nugget contains a globally unique id that is generated by fingerprinting the basic nugget content, so looking for duplicates is a matter of checking for items in a node.

Once all the primary routes have been sent, the component switches to relaying to secondary routes, then to tertiary, etc. until the search reaches its expiration time or hop count, or the route table is exhausted. When the search expires (either by a timeout or hop count condition), the random pubsub node is destroyed and future publishes will not get routed to that node's subscribers.

The algorithm for choosing the ordering of remote components to relay searches to is based on the order of where search success is likely. Future work could be done here to optimize this algorithm, for real world scenarios could suggest alterations. For now however, we relay searches based on the following ordering of likelihoods for successful searches:

1. Past successful "hits" for the same search criteria.

2. LLI servers whose primary bounding box (or "sweet spot") contains our search location bounds.

3. LLI servers whose Nth bounding box contains our search location bounds. N { 2 … total route bounds –1 }

4. LLI servers whose last bounding box (or "the extents") contains our search location.

5. A random server from our route table not already visited in 1-4. This server should then attempt to relay to others on its route table using similar criteria.

LLI Network Awareness

Being aware of the network means both being a part of it and learning as much as we can from it. LLI servers are *motivated* to become experts on a particular area of the globe.

LLI servers maintain a routing table of other LLI servers. This table includes both presence information (online, offline, too busy), as well as various bounding boxes for statistical density concentrations for the content they manage. Servers use this information to send search requests to the servers most likely to contain an answer to search requests. Searches only occur on LLI servers that are in online state and whose bounding box contains the search area in question.

The following pseudo-code outlines part of what LLI servers do to maintain awareness:

```
# Make sure we have a route table
if (size(@routes) == 0)
{
    # we have an empty route table, prime it from another host’s past
experience
    prime_route_table($config_file{ route_primer_host });
}

# get the current presence of our route table entries
foreach $route (@routes)
{
      $presence = get_presence($route);
      update_route_presence($presence);
}

# Snoop on incoming search requests
$search_request = get_new_search();
if (in_route_table($search_request->lli_component) == FALSE)
{
      if (route_table_full == FALSE)
      {
         add_route($search_request->lli_component);
      }
}

# Becoming an expert on each of our routes, by requesting notification
add_route ($component_jid):
{
  …

      $my_expert_cell = get_my_expert_area();
      subscribe_remote_node($component_jid, $my_expert_cell);
  …
}
```

In addition to the above, the LLI component must unsubscribe LLI servers that request to be unsubscribed from its route, following the "social rules" of the network.

Storing Location Nuggets

While searching for location-keyed information is an important function of the LLI server, it is a meaningless function without data on which to search. Thus, the other side of search is data storage, which LLI servers will do for jids that match their storage criteria. Like other parts of the LLI system, a storage event is generated when a XML packet that matches our storage namespace is received. We can think of the storage system in LLI as being composed of three layers: the access control, pubsub, and storage backend.

While the namespace for location nuggets is shared amongst all LLI servers a jid will typically publish to his home server, rather than to a "foreign" storage jid. Because bits are cheap, but not completely free, usually LLI servers will only accept location nuggets for storage from "their own" jids. This property is not mandated however, and LLI servers could be configured without access control at all, should a LLI server operator desire. Similarly, because not all location nuggets are public, a jid might want to trust his home server not to distribute nuggets unless they meet the distribution criteria as set forth by the owner jid. While it would be pleasant to be able to publish anywhere and trust that distribution desires are honored, the infrastructure does not enforce this, so there should be no expectation of data privacy from untrustworthy LLI servers.

Once a "storage" packet gets past the LLI access control sentry layer, it gets passed into the pubsub layer that determines the correct node to associate the data with, as well as enforces pubsub-level access and data typing. While we examine the LLI storage system in three layers, conceptually it is first and foremost a pubsub storage system. LLI utilizes the pubsub protocol (as defined in JEP-0060 [40]) as the channel for all storage calls. By wrapping extra, application specific, logic around the inner "data bucket" we are able to tailor the functionality to meet our needs. The LLI-specific logic breaks apart the XML-based data nuggets to examine their content's validity and distribution scope.

Once the pubsub layer finishes verifying validity of the pubsub node name and access control, it passes the parsed data off to a data storage backend. The technique that LLI servers chose for data storage is implementation-specific and depends on the goals of the LLI server. Backend storage could be an SQL database, flat file, XML file, or any other appropriate scheme. Once the storage packet's data reaches the backend database and is successfully archived, the LLI server's location statistics and bounding boxes should be updated, although for efficiency reasons this could happen every Nth storage or M seconds.

Relaying Nuggets

As mentioned in the storage section above, the location nuggets are stored within a slightly specialized pubsub scheme. As such, the entry of new nuggets into the LLI system could be thought of as an event that is thrown and caught by the pubsub's data distribution subsystem. Before going much further we should briefly examine the affiliation concept as viewed by the pubsub system.

The following table from JEP-0060 lists the various pubsub functions and affiliation rights.

Table 4-1. PubSub Node Affiliations

Affiliation	Subscribe	Publish Items	Purge Items	Configure Node	Delete Node	Delete Item
Owner	🟩	🟩	🟩	🟩	🟩	🟩
Publisher	🟩	YES	🟧	🟧	🟧	Publisher
None	🟩	🟧	🟧	NO	🟧	🟧
Outcast	🟧	🟧	🟧	🟧	🟧	🟧

In our implementation of pubsub, the LLI component manages each of these rights, based on both node-owner desires (in cases such as "I want to subscribe to your location node") as well as in system-wide rules (in cases such as "no, you can not delete a cell node, that is owned by the component"). Also note that in the table the "items" are what we refer to as "nuggets." Pubsub components contain nodes which contain both sub-nodes and data items. Nodes have owners, publishers, subscribers, and outcasts associated with them. Data items are contained in a node and may carry a payload with content in it.

Relaying of location nuggets then is going through the subscriber list of jids for a particular node that a nugget belongs to and sending out pubsub-formatted XML containing our location nuggets to each subscriber.

Becoming an Expert

Since most of the time data published to a particular LLI server will be from a finite set of users, and since it is also likely that these users will share either a general, boundable geographic location (such as at a university or ISP) or a common set of interests (such as searching belonging to the same company or sharing a hobby of deep sea diving), one could assume that each LLI server will have a natural clumping of data. This clumping could be that most data points are likely to be over land, or 90% of the data points are in Europe, or all of the data points are within 15 miles of Poughkeepsie, New York. Regardless of the semantic phrasing of the clustering, it is assumed that the data of each particular LLI server is not a random distribution.

Assuming there will be a natural bounding to a LLI server's data lets us take some liberties in designing optimization strategies for answering searches made by a LLI server's users. If we can cache location nuggets that are within the area that our users likely will go, we can provide shorter search times because we can relay more information from local sources rather than having to depend on a remote search. This strategy is thought of as "becoming an expert" on an area, which technically condenses into having a tunable cache based on location.

LLI servers become experts by watching the location nugget traffic that other users and LLI servers pass through its server. When a public nugget that meets our expert area criteria is discovered, that nugget is plucked out and stored in our cache.

Proxies Payment

Location nuggets contain pointers to other content. This extra content could be restricted and require a payment in order for viewing to occur. Clients might be made aware that payment is required by receiving an HTTP error code of 402, after which clients could ask their home components for help. The current computational climate suggests that client devices might not have the data storage, processing power, or network bandwidth to reliably negotiate location-based content. Furthermore, content prices could be too high to support a lack of certifiable quality before purchase— said another way, the current economic infrastructure favors larger content producers with brand-name recognition who can convince consumers to pay the $4 minimum/transaction that is required to make a profit. Micro-payment schemes have been touted as a way for lower priced transactions to occur which carry less risk to the consumer and still economically induce those producers with relevant information to make it available.

This function of the LLI server is optional and not prototyped in this version, but as jids connect to a trusted server, that server could be used as a secure payment proxy for the jid. By strengthening the already implicit trust relationship between jids and LLI servers, rules such as "allow me to spend $5/day on content before warning me," or “purchase content for me if it is less than $0.005/viewing” or “only purchase content with a good rating” could be implemented. LLI-agents might base these rules for payment on existing web-based payment agents such as PayTrust or Bills.com [8].

Cleans Up

The asynchronous nature of searches and the LLI system in general causes data to accumulate, waiting for someone to either request the information, or for search answers to be returned. Because of this property, LLI servers need to actively clean up their data from time to time, “garbage collecting” expired location nuggets, or nodes whose timeout value has been reached. Similarly, from time to time a LLI server’s location nugget statistics need to be recalculated with the most accurate/up to date coordinates. Other expirations of node-subscription properties may also be required, and need to be updated periodically. While not a primary functionality of Location Linked Information, it is crucial to data accuracy and storage efficiency of the server component.

Maintaining Node Access Control

As mentioned previously in this section, the LLI server component is a modified pubsub implementation that has extra logic wrapped around the data format and which implements a search function. As such, one of the primary functions of the component is to manage node affiliation and access control requests.

There are three root node hierarchies in the LLI system: generic/, presence/, and cell/, each of which has its own access control rules and logic.

Generic nodes (generic/) are created as a byproduct of search requests, and as such are owned by the LLI component. The LLI component then has the sole authorization to delete the node, as well as to purge items. Since generic nodes are used to relay search results to a searcher; it has an open subscriber and publisher policy, so that any interested party can subscribe and publish to the node, although in practice this is unlikely to occur, as generic node names are not correlated to a jid or particular search string. LLI implementations may choose different algorithms for naming generic nodes (such as using uuids), however the prototype implementation names generic nodes by creating a random string of alphanumeric characters. End user clients will not create generic nodes directly, but do so indirectly through initiating a search request that auto-vivifies the generic node and returns it to the client.

Presence nodes (presence/) are for storing location data nuggets that are directly related to a particular jid. For instance, presence/ yohan@jabber.media.mit.edu would be the node for storing location nuggets pertaining to yohan@jabber.media.mit.edu. The LLI component may wish to limit the jids that it stores presence information for. A logical access control policy for these nodes might be to only allow nodes in the presence/ hierarchy for jids that are considered "local".

Publishing to presence/ nodes requires that the sub node name (the part after the initial presence/) match the base jid (that is, the jid without the resource: yohan@jabber.media.mit.edu, not yohan@jabber.media.mit.edu/Work) of the publisher. The effect of this is that the presence/ hierarchy is used to store the current location information about your own jid rather than someone else's. Remember that the routing of jids is controlled by the Jabber protocol that takes steps to verify that the jid in the "from" or "to" fields of the top level XML tags (<message/>, <iq/>, or <presence/>), so we have some level of assurance that the information is valid or at least from a trusted source.

Finally, subscribers to the presence/ hierarchy are controlled based on specific "subscription requests." These subscription requests are essentially asking, "I would like to be notified of changes to your location" to a particular jid. When presence/ nodes receive subscription requests, they relay them to the node owner (the jid), which can then accept, accept with a timeout, or deny the subscription request. If the subscription request is accepted, future publishes will be relayed to that jid.

A modification of the "accepted" type of subscription request is the subscription request accepted with a timeout value. The timed out subscription allows us to grant the right to our location data for a finite time period, making the subscription revert to "unsubscribed" when the validity period has been passed. This feature is important for mirroring real-world situations where we might want to allow someone to find us for a limited period of time, such as while at a convention, but would not want our entire location data stream sent to the authorized jid in perpetuity.

Like generic/ type nodes, presence/ nodes are auto-vivified. On the first publish to a node that meets the reflexive access control requirements for that node, the node is created if it does not already exist. This property reduces the required XML sent to the node for a state-less client that may or may not remember if she has ever published an item to their presence/ node before or not.

Cell nodes. The final node type is the cell/ hierarchy, responsible for storing location nuggets that are not tied to a particular jid entity. The specifications of the naming of items within the cell hierarchy, as well as the node names themselves, are slightly more complicated, however the access control restrictions are easily explained. Cell/ nodes are owned by the LLI component, and thus are not directly purge/delete/create-able, by external jids. Cell nodes have an open subscription policy, allowing any requesting jid to be subscribed to any cell node. Publishing to cell-type nodes should be restricted to only local jids, as this further isolates the unscrupulous location-based spammer to polluting his own machine.

How are cell nodes and items within them named? Cells are inspired from the Digital Earth/GeoWeb project [14], which split the Earth up into a finite number of cells that could then be directly addressed. The way this works is first latitude and longitude expressed in degrees decimal are converted to the sexagesimal format of DD:MM:SS (degrees, minutes, seconds), so -71.234 gets converted to -71:14:02. The latitude and longitude then get sequentialized into a name, as follows:

```
cell name   =
10Dlat,10Dlon:1Dlat,1Dlon:Mlat,Mlon:Slat,Slon
            =     -70,150:1,9:14,1:2,59
            =     (-71:14:02, 159:1:59)
```

Where 10Dlat means the 10s unit of the latitude, 1Dlon means the ones unit of the longitude, Mlat is the minutes of latitude, and Slon the seconds of longitude. The effect of this scheme is that we can address 1 second X 1 second cells on the earth, which depending on the position on the earth is about 27 meters x 27 meters. The hierarchy delimiter is the colon; leaving off precision will yield larger cell sizes, for instance, -70,150.

Items (nuggets) stored within the cell will have a greater precision for the latitude and longitude, allowing exact positioning, including an altitude component. Item names are created by using the SHA1 hash algorithm, which creates a unique output for a given input, on the concatenation of (latitude, longitude, type, source, URI, altitude, expiration). Consequently, if any of these components changes, the nugget is considered to be a different nugget. Similarly, if all of these constituent parts are the same, two nuggets are considered identical.

Since cell nodes are more general than presence nodes, and the data stored in presence nodes is expressed in longitude and latitude coordinates that can be converted to the cell node-naming scheme, why have presence nodes at all? The answer lies in the gulf between the access permissions of each node type. We want to have stricter access restrictions on entity location and state than on location-keyed data. Also this division allows us to codify the differences previously mentioned between the concepts of "location" and "presence". Presence, remember, is from the perspective of the container ("Yohan is present in his office.") whereas location is from the perspective of the entity ("I am in Cambridge, Massachusetts.")

LLI Component Technical Details

Thus far we have covered the basic rules that define the inner workings of the LLI server-side Jabber component, but have left out many of the technical details employed to build the server prototype. This section will fill in some of the gaps in the LLI server implementation, however these details should be taken as secondary to the conventions set forth above, for real-world experience will likely expose numerous areas where optimizations to this prototype are desirable.

Remember that the LLI server is actually a pluggable component of a Jabber server. The Jabber server chosen for the prototype was the open source jabberd 1.4, [34] which is the first stable open source Jabber server. Connecting components to jabberd is done by utilizing one of three methods: loading in via a shared object (.so) library, connecting via TCP sockets, or executing directly by jabberd and communicating via Standard I/O (STDIO) [2]. The prototype chose to connect to jabberd via TCP sockets. Utilizing TCP sockets for the jabberd connection allows us to have the LLI component on either a different physical machine or run as a different user, should this be desired. Components connect to jabberd via XML streams, just as clients and servers are connected via an XML stream.

Written in Perl and utilizing the Net::Jabber and XML::Twig Perl libraries, the LLI component consists of approximately 9,000 lines of code. As a backend storage subsystem, the component uses the MySQL database through the abstract DBI Perl interface. The main LLI component lives within a single process and is not multi-threaded. The LLI component follows an event-driven, callback style, construction, reading from the XML stream given to it by the jabberd. When the LLI component sees an XML fragment within the Jabber stream that it recognizes it passes that fragment to a callback handler that takes some action on the data.

To accomplish some parallelization, the searching sub-system is split into a second process that shares database access with the main component. This searcher component, ('lli-searcher ") is awoken from its sleep cycle by the main LLI component whenever the component receives a search request. The searcher then carries out the local and remote searches and relays the results to the requesting jid.

LLI DATA

When talking about the data in LLI, it is useful to distinguish between the three types: LLI nugget data, LLI glean data, and LLI referenced data. Whereas LLI nugget and glean data resides within the LLI system and is accessible via LLI server components, LLI referenced data resides at a URI. It is the nugget data which links together the location (latitude, longitude, altitude) and reference URI. The glean data then provides a popular vote for the validity of the nugget data.

LLI Nugget Data

Nugget data is the basic content container for LLI. It contains a globally unique identifier, data type, latitude, longitude, altitude, expiration time, and URI. Other fields may be added, such as cost or textual description, but they are not required. Latitude and longitude are standardized in decimal degrees, using the World Geodetic System 1984 (WGS 1984) datum which specifies a global reference and coordinate system. Altitude is in meters. Expiration time is in seconds since the epoch (00:00:00 January 1, 1970), using UTC.

LLI Glean Data

Glean space is the communal forum for ranking nugget data. Glean space is open, with anyone allowed to annotate the validity of a nugget. Clients and servers then assemble any of these left behind bits to piece together a picture of the true nature of the nugget— Is it annoying? Was it typed correctly (it claimed to be a review, but it was really an advertisement)? Was it worth the money? Derived from the practice of allowing the public to pick through and gather left-behind crops after they had been harvested, creating a legal quasi-public space from private lands.

It is intended that glean data cannot be deleted; only amended with another annotation. The only required field for a glean nugget is the nugget-id that it refers to. Any other field will be interpreted as appropriate by the client or server. For instance, a client might add a "rating" field to the glean nugget and give it a negative value. Servers could then be configured to tally up all the rating fields, and deny transmitting the nugget to the client if its value is below a certain threshold. Similarly, one might annotate a nugget with a text description that might get passed to a client for interpretation. Servers could be told to ignore the nugget-data given "type" if a certain number of glean nuggets suggest that the actual type is something else.

This division between content and social worthiness allows the content creators control, while motivating them to manufacture socially meaningful data, knowing that if they do not it will get moderated into the information sea, never to be found, or better yet, corrected to the true value. Glean space differs from the wiki concept [61] where content is given to the public to manage. Experience with the wiki has shown that strict version control needs to be implemented in case some of the public's changes are incorrect. With glean nuggets, instead of content being directly turned over to the public, an open opinion and annotation layer is added, and combined at run-time when the content is presented.

LLI Referenced Data

LLI referenced data may be of any type that can be referenced by a URI— plain text, HTML web page, image, audio, video, etc. While any type of data may be referenced, it can be imagined how it would be useful to have the URI link first to a definition of a space, such as would be made possible by linking to an RDF [48] description, which would then point to other data containers. It is surmised that given an implementation of such a location linked system, links would first be made directly to existing data, but over time the practice of referencing metadata would take over, for future generations of applications can make use of this extra information.

LLI Data Streams

Gluing LLI server components together with other LLI servers and LLI clients is an XML data stream enveloped in a Jabber XML stream. Further information about the Jabber encoding can be found in either IETF XMPP specifications or at the central repository for Jabber protocol information, the jabber.org website.

LLI Clients

Whereas LLI server components actually carry out the data storage, searching, and information disbursal tasks, humans and other "end users" utilize LLI clients to view the information stored by LLI's server side. It is worth reiterating that the only requirement for an LLI client or server is that they "speak" a compatible XML stream, as defined in the XML stream section to follow. While any number of optimizations and variations could be made on the LLI component, it would be expected that the LLI client side would have an even wider variation amongst purposes, form-factors, and task-specific usability.

It is possible that LLI clients take many forms: as desktop application, digital bot, cellular telephone, specialized operating system, embedded within handheld devices such as dynamic maps, or in any other as-of-yet uninstantiated devices. Indeed this potential for multiple implementations, and the possibility for the iteration of the client-side design is the prime reason for choosing a protocol-based infrastructure. By defining the channel interface it allows the designer to optimize the endpoints for a specific task, or to iterate on the design and maintain backwards compatibility that becomes important as the technology matures and moves from the computational timeframe to the urban scale.

For this study I created a general-purpose LLI client prototype called Janus as well as a few bots that insert data into an LLI component from external sensors.

Janus

In Roman mythology, Janus was the god of gates and doorways, with two faces looking in opposite directions. I think of Janus as looking between virtual and physical space, providing a gateway between the two. Janus was developed as a "desktop" style application, meant to run under the Linux operating system, although as it is written in the C programming language, and depends on the GTK+ library, so it is theoretically possible to compile under different operating systems, such as Mac OS or Microsoft Windows, although I have not tried this exercise as of this writing [27].

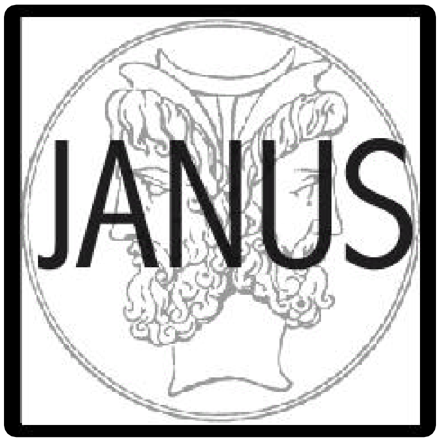


Figure 4-4. The Roman god Janus.

Janus, the dynamic map viewer, assembles a map of your surroundings that includes all the location nuggets in your vicinity that match a user-defined search criterion. Janus knows its current position, expressed in latitude, longitude, and altitude, and maintains a connection to its home LLI server, or more accurately its home Jabber server that has an LLI component on it. These two conditions, a) knowing the current location, and b) being connected to the home Jabber server, would seem to necessitate a constant connection to a GPS and the Internet. While having both would be ideal, there are clever ways to work around a), and the loss of b) should not damage the current read-out. This kind of multi-faceted robustness is required when we move from the expectedly fragile desktop to solid environment, and is thought to be a formidable obstacle for future generations of interface designers.

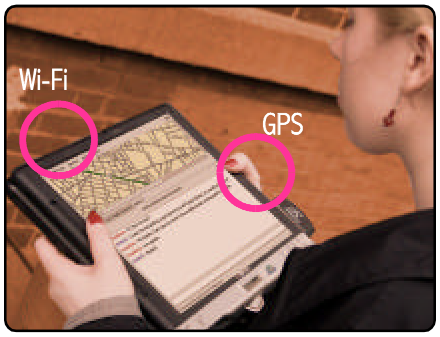
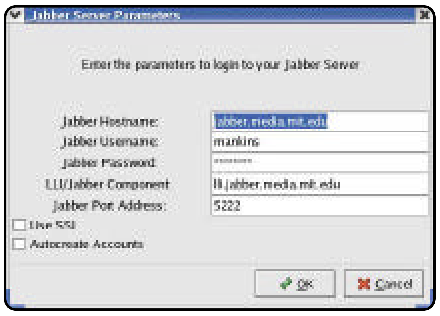


Figure 4-5. The prototyped Janus client was in a Tablet PC and contained both GPS and Wi-Fi.

The Janus Tour

Rather than list each of the features of the client and explain them piece-by-piece, we will take a screenshot tour of a typical client session, outlining some of the peculiar features that a location-client might require.



Enter the parameters to login to your Jabber Server

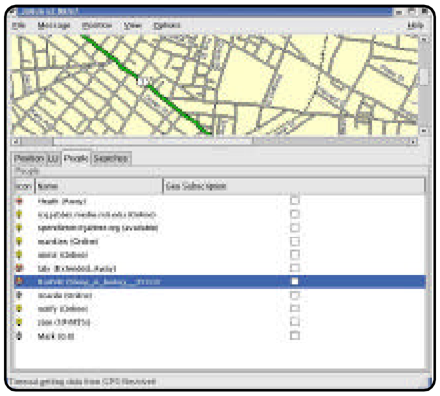
Jabber Hostname:	jabber.media.mit.edu
Jabber Username:	mankins
Jabber Password:	********
LLI/Jabber Component:	lli.jabber.media.mit.edu
Jabber Port Address:	5222

☐ Use SSL
☐ Autocreate Accounts

Figure 4-6. Login to the home Jabber server.

The first thing you will notice when starting up the Janus client, assuming it is the first time you have used the client, is that it will ask for a Jabber login, hostname, password, and a few other jabber-centric parameters. It would certainly be desirous to eliminate all such login and text-box entry related nonsense, however the benefits of trusted client-server relationship outweigh the kludgey user interface. I think this caveat is more meaningful if you consider that one of the design inspirations of the Janus software was the map. This will require a brief diversion into the motivations for creating the LLI client, however the insight should prove useful throughout the rest of the tour.

Janus: The Digital, Dynamic Map



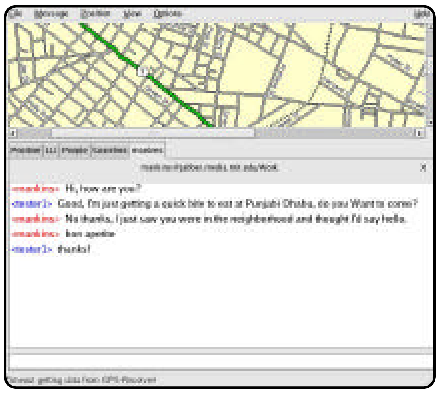
Heath (Away)
[illegible]
[illegible]
mankins (Online)
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
Mark [illegible]

Timeout getting data from GPS Receiver

Figure 4-7. The roster of nearby friends.

Janus wants to be a digital, dynamic map. In a perfect world where software designers can throw their ideas into "iteration machines" which iterate through all the bad implementations and pull out a good one, Janus would not just aspire to be a map, but it would be a paper-thin map— the kind of thing that you fold up in your back pocket and pull out from time to time as you wind your way through the city's terrain. The difference between the traditional, paper-thin kind of map and Janus, of course, would be that Janus would present a dynamic view of the city, with the omnipresent "you are here" clearly marked. Not only would Janus have a "you are here" that constantly gets updated, but it would have similar markers for other people and mobile entities (cars, motorcycles, trains, buses, subways, airplanes, boats, pets, police vehicles, segways, etc.) that are LLI-enabled.

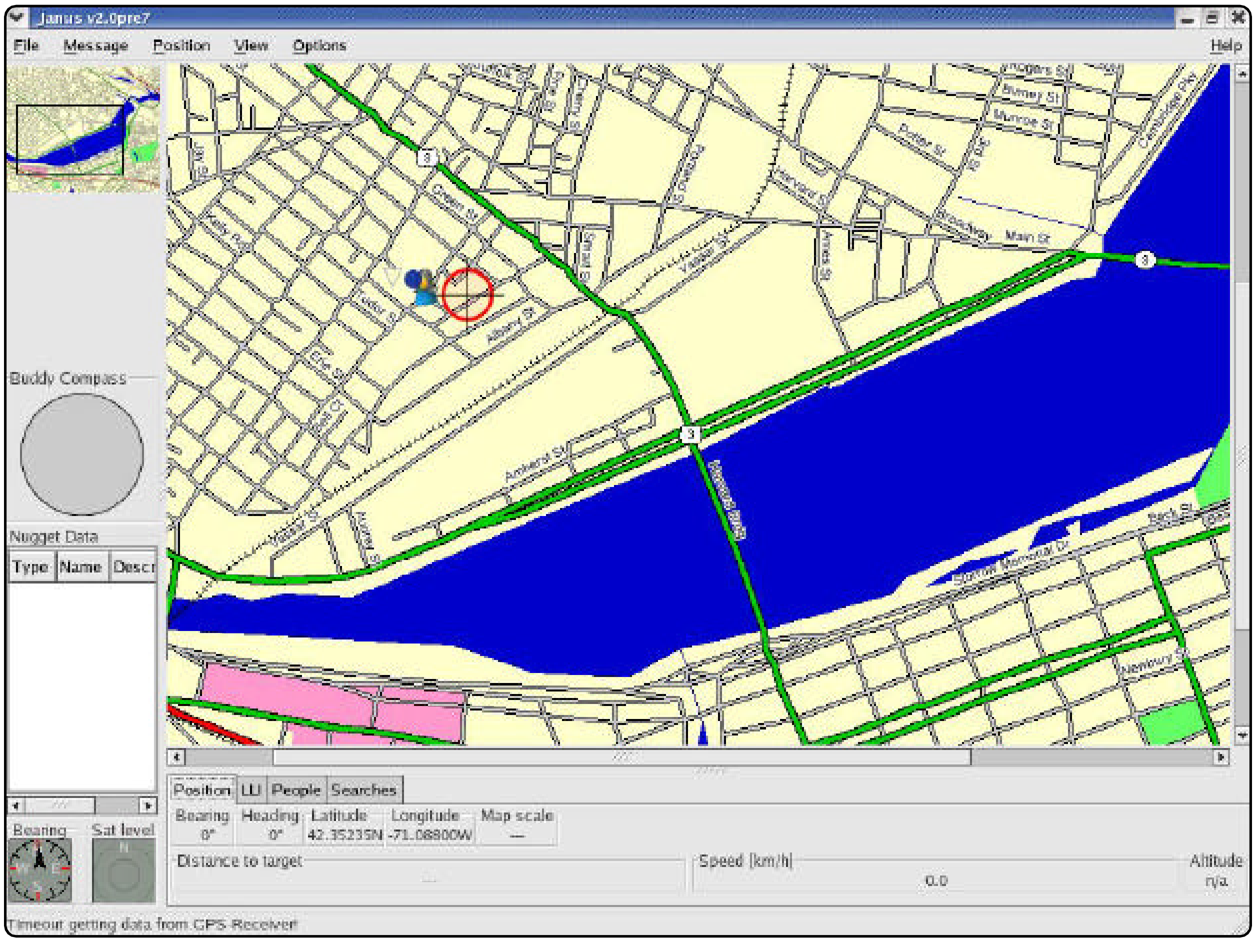
Of course these types of dynamic maps already exist in the in-car navigation domain, and they are quite good at helping to wayfind and even acquaint you with your surroundings. So what makes Janus different from an in-car navigation system? Well, two things: a) the information contained in Janus is entered from the bottom-up, by a thousand other people like yourself who were kind enough to leave a piece of knowledge behind for others to benefit— in car navigation is typically a stagnant, top-down approach that takes years to accumulate and is often out-dated; and b) Janus has a user interface that is designed for annotating space as well as viewing other's spatial annotations. Janus' UI is more of a portal for communicating with the city— kind of like a scratch and sniff map where you get something back from the interaction.



File
Message
Position
View
Options
Help
Position
LLI
People
Searches
markins
markins@jabber.media.mit.edu/Work
Timeout getting data from GPS Receiver

<markins> Hi, how are you?
<tester1> Good, I'm just getting a quick bite to eat at Punjabi Dhaba, do you Want to come?
<markins> No thanks, I just saw you were in the neighborhood and thought I'd say hello.
<markins> bon apetite
<tester1> thanks!

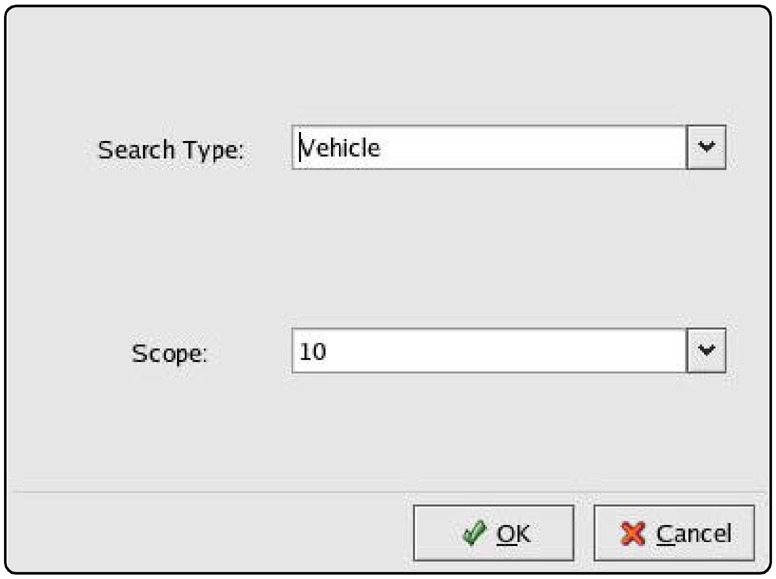
Figure 4-8. Chatting with others within the Janus client.



Bearing	Heading	Latitude	Longitude	Map scale
0°	0°	42.35235N	-71.08800W	—

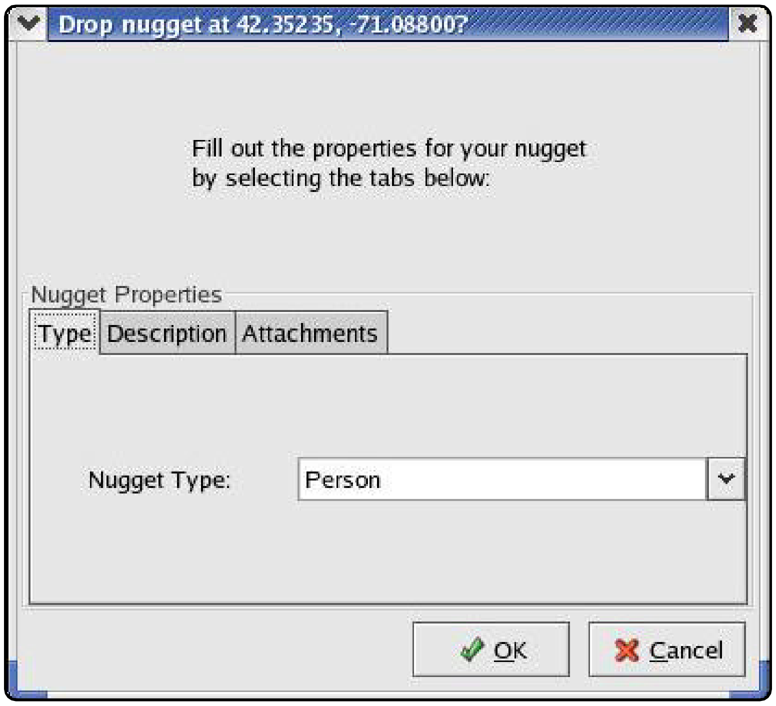
Distance to target: ---
Speed [km/h]: 0.0
Altitude: n/a

Figure 4-9. This is the "main" screen of the Janus application. The map display is the predominate interface element. A red cross marks "you are here." In this screenshot, a person icon is shown, along with other nugget icons that can be accessed by pointing.


Search Type: Vehicle

Scope: 10

Figure 4-10. The current search interface is basic: specify the type of nugget and a search radius in which to look.

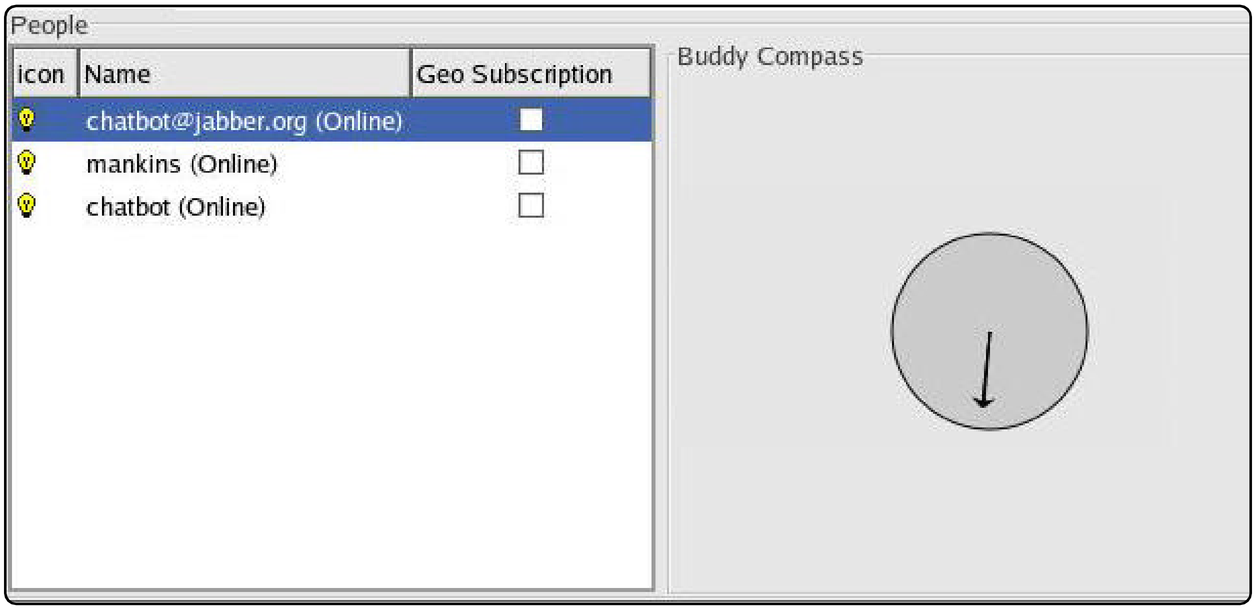

Drop nugget at 42.35235, -71.088007?

Fill out the properties for your nugget by selecting the tabs below:

Nugget Properties

Nugget Type: Person

Figure 4-11. Leaving a nugget annotation involves setting the type, leaving a basic description, and linking to a URI.


People

icon	Name	Geo Subscription
	chatbot@jabber.org (Online)	☐
	mankins (Online)	☐
	chatbot (Online)	☐

Figure 4-12. Select a roster item, and your buddy compass will show you which direction that person is located.

LLI: What It's Not

So far we have been discussing what LLI is, but what is it not intended for? Firstly, LLI was not intended to directly transfer three-dimensional vector data, such as would be found in a GIS database or CAD file. The complexities of three-dimensional spaces balloon the amount of data required so that it is not practical on current wireless links and within the current Jabber server environment. Instead, LLI is meant for storing points in three-dimensional space that then reference other documents. These other documents could very well be CAD files or GIS layers, however the data itself is not stored at the LLI level.

CHAPTER FIVE: PUBLIC COMPUTING

"The problem is to find a form of association which will defend and protect with the whole common force the person and goods of each associate, and in which each, while uniting himself with all, may still obey himself alone, and remain as free as before."
–Jean-Jacques Rousseau

Chapter one suggested that bringing digital technology into the city, specifically into the public realm, would increase the city's potential. Set against this criterion, we can evaluate the effect a technology such as LLI might have on increasing urban livability. In thinking about a world where LLI is deployed, it becomes evident that the challenge of maximizing urban potential in the Smart City is complicated by the sometimes-divergent goals of the western city's inhabitants: citizens (maximize wealth), government (minimize risk), and digital life (maximize data).

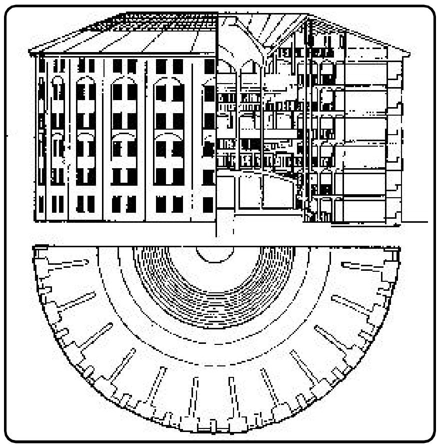


Figure 5-1. Plan for the *Panopticon* by Jeremy Bentham. Iconic of a watched society, the Panopticon was studied by Foucault who found it embodied the essence of power with its asymmetrical division of knowledge, and thus power. Some of the worries for location systems is that the technology creates a digital Panopticon.

Moving further in time, toward the Smart City, the once-clear demarcation between the virtual and the real worlds becomes increasingly difficult to visualize, and the challenge for designers is avoiding the urban panoptic state of constant surveillance. While it seems plausible that LLI, when deployed, would meet with general acceptance as a functional technology, the social ramifications of creating a "watched" society merit further exploration. Indeed, in thinking about possible future scenarios, it is easy to imagine a world where "government" and "databases" become too powerful, as they silently record the movements of the citizenry. To rebut these charges, this chapter introduces "public computing," a framework for designers of technology like LLI.

The Missing Zone of Digital Interaction: Public

Anthropologist Edward Hall used space to classify scales of human interaction into *intimate* (0 – 46cm), *personal* (46 – 122cm), *social* (122 – 304cm), and *public* (304cm – infinity) [30]. While Hall was interested in cross-cultural differences in the use of personal space, his taxonomy proves useful today for analyzing computer-human interaction. Analogues of Hall's zones exist in the language of technology: his intimate zone is *wearable computing*, the personal zone is *personal computing*, the social zone is *proximity computing*, and for the emerging public zone I suggest the term *public computing*.

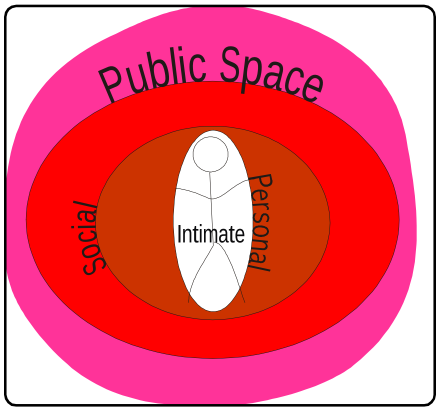


Figure 5-2. Edward Hall's classification of space. Intimate zone is wearable computing, personal is personal computing, social is proximity computing, and for public I suggest public computing.

The Fear of the Public Scale: Big Brother

What is so different about the public scale that it needs special attention? The short answer to this question is that not enough research has been conducted at the scale of the public to know how computing and the public will interplay, and as a result dystopian visions seem just as likely as Utopian ones. Will networked sensors become the embodiment of Big Brother, or will increased connectivity lead us to a second Pax Romana?

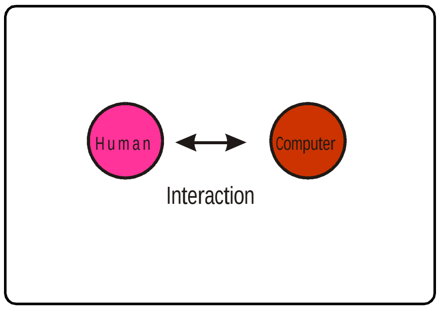


Figure 5-3. The HCI model has the human and computer directly interacting with one another. Contrast this to the HCEI (Human Computer Environment Interaction) model, where both computers and humans interact directly with the environment.

Public Computing: A Direct Interface to Society

Answering questions on the use of technology in public will require communal discourse, numerous generations of public-scaled digital devices, and a set of precepts that describes appropriate social practices. In general, however, it is sufficient to say that public computing provides a direct interface to society, adding a potential for significant benefits as well as risks.

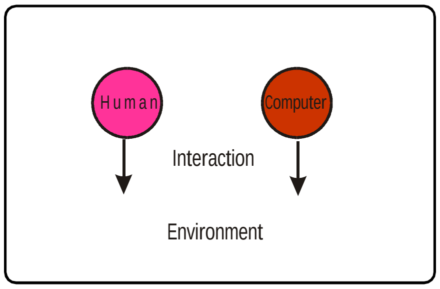


Figure 5-4. Human Computer Environment Interaction model.

The nearest recent event to such a widespread and rapid infiltration of digital technology into society was the rise of the Internet and WWW. These technologies are continuing to serve as catalysts for change, sparking questions of traditions, challenging laws, and probing the roles between institutions and individuals. It can only be surmised that pushing these technologies into the realm of the public will multiply these trends. We now explore some of the properties that describe public computing.

PUBLIC COMPUTING: DESCRIBED

Public computing integrates digital devices, storage, and networks with Hall's spatial classification of *public space*, the economic *public good,* and notions of the *public discourse*. These three "other" publics— space, goods, and discourse— provide clues to the ways public computing might be different from other computational paradigms.

The first, Hall's *public space, concerns an interaction of objects at a distance*, and implies that spatial relationships will be used to provide cues as to the kinds of activities that occur in an area. In this view, space becomes the portal to the virtual world, providing a context and setting for information. Other paradigms view the technology as being the primary gateway to information— the desktop is the intermediary for programs and their functions. While technology still provides the gateway to information in public computing, it only gives a path to *some* information, and not necessarily *all* information.

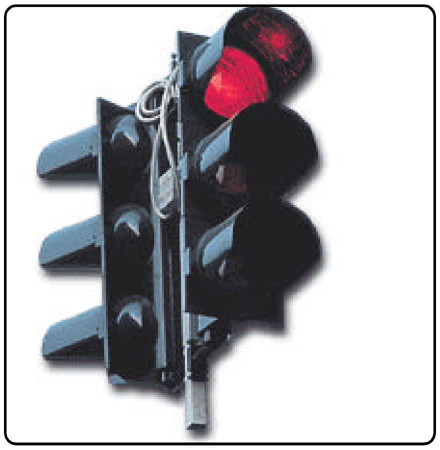


Figure 5-5. Traditional public goods have been stationary and physical, like the traffic signal. With public computing, it is the computation and data that is in the public sphere, but the principles are the same.

Economists define the second component, the public good, as a good that is *non-exclusionary* and *non-rival* in consumption [36]. Said another way, *public goods can be used by anyone as soon as they are made available and can be used by multiple people simultaneously*. A traffic signal is an example of a public good— it will not be exhausted when multiple people look at it simultaneously, and it would be inefficient (and strange) to allow one person to look at it while blocking others from partaking in its use; another classic example of a public good is clean air. Public computing relies on being connected to the Internet, and piggybacks on this infrastructure to enable multiple people to have simultaneous and unlimited access to digital knowledge.

The final component to public computing is the promotion of open discourse and the notion of *"best answer wins."* In his *Politics,* Aristotle examines this kind of active participation in public discussions in order to determine the laws of the land as being healthy for society, for active participation taps into the collective wisdom and gives a voice to the disenfranchised. The Internet and digital technologies allow a new scale of public opinion to be practical, improving both the quantity and time span in which beliefs are collected, thus improving the *aggregate knowledge* of society. Desktop computing had little link with society except through its human users who, by proxy, were responsible for any actions that take place within the virtual realm. In public computing, the interface to society is direct, and clear expectations for the rights and responsibilities of the human and digital inhabitants are crucial to the livability of the Smart City.

The Modified Social Contract

In the United States, broad statements in the Constitution and Bill of Rights set forth the basis by which society functions. I suggest that we augment these rules to acknowledge that we are living alongside digital organisms. As these digital organisms become increasingly sophisticated, a new series of guidelines will be required to judge their behavior against, insuring that digital life "plays nicely" in our society.

One way in which this might be done is to create a "social contract" between human and digital, in which some human rights, such as the right to privacy, are surrendered in return for increased connectedness and a better quality of life.

So those implementing technologies in the public computing zone should:

1. Pay attention to the spatial considerations and interfaces that are unique to public space.

2. Keep technology accessible to all; it also should be as extensible and compatible as possible and should be made to last in the tens of years.

3. Be cognizant that others have the freedom to overlay and create digital information, residing in the public domain.

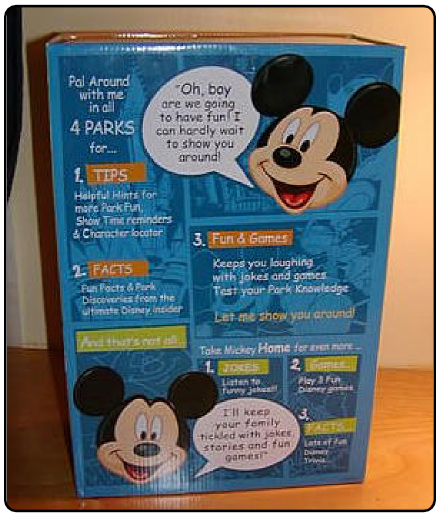

Pal Around with me in all 4 PARKS for...
"Oh, boy are we going to have fun! I can hardly wait to show you around!
1. TIPS
Helpful Hints for more Park Fun, Show Time reminders & Character locator
2. FACTS
Fun Facts & Park Discoveries from the ultimate Disney insider
3. Fun & Games
Keeps you laughing with jokes and games Test your Park Knowledge
Let me show you around!
And that's not all...
Take Mickey Home for even more...
1. JOKES
Listen to funny jokes!!
2. Games
3. FACTS
I'll keep your family tickled with jokes, stories and fun games!"

Figure 5-6. This product by Walt Disney follows you around as you tour the Magic Kingdom, giving you a "personal tour" by a toy Mickey Mouse. Because this implementation is a closed system, which has not agreed to be bound by any social rules, there exists an opportunity to abuse the data stream. Already it has been reported that the Walt Disney Company has analyzed the data and found that after buying the item, most people go to find the "real" Mickey Mouse. The question that has people worried is "What next?"

In return, the digital agents need to be made aware of their responsibilities:

1. Information owned by other entities, be they digital or human, is considered private, and must not be shared with others unless explicitly granted the rights to do so.

2. Identity forgery and unauthorized data disclosure are forbidden, socially unacceptable activities, and must not occur under any circumstances.

REFERENCES

[1] Abowd, G, et al. "Cyberguide: A mobile context-aware tour guide." ACM Wireless Networks 3 (1997): 421-433.

[2] Adams, D J. Programming Jabber. Cambridge: O Reilly & Associates, 2002.

[3] Aguilera, Marcos K., et al. "Matching Events in a Content-Based Subscription System." Symposium on Principles of Distributed Computing (1999): 53-61. <http://www.research.ibm.com/gryphon/Gryphon/>.

[4] Amsterdam Realtime. Waag Society. <http://www.waag.org/realtime>.

[5] Ayatsuka, Y, K Hayashi, and J Rekimoto. "Augment-able reality: Situated communications through physical and digital spaces." Proceedings of the 2 nd International Symposium on Wearable Computers (1998): 68-75.

[6] Azuma, R. "A survey of augmented reality." SIGGRAPH 95 Proceedings, Course Notes #9: Developing Advanced Virtual Reality Applications (1995): 1-38.

[7] Big Brother System and Network Monitor. <http://bb4.com/>.

[8] Bills.com Online Bill Pay. <http://www.bills.com/>.

[9] Brown, Denise, Steven Izenour, and R Venturi. Learning from Las Vegas. Cambridge: MIT P, 1972.

[10] Carzaniga, Antonio, David S. Rosenblum, and Alexander L. Wolf. "Achieving scalability and expressiveness in an Internet-scale event notification service." Symposium on Principles of Distributed Computing (2000): 219-227. <http://www.cs.colorado.edu/users/carzanig/siena/>.

[11] Cheverst, Keith, et al. "Experiences of Developing and Deploying a Context-Aware Tourist Guide: The GUIDE Project." Proceedings of MobiCom (2000). <http://www.comp.lancs.ac.uk/computing/users/kc/Papers/mobicom.pdf>.

[12] Christmas Bird Count. Audubon Society. <http://www.audubon.org/bird/cbc/>.

[13] Cooltown. HP Labs. <http://www.cooltown.com/cooltownhome/>.

[14] Digital Earth: Building the New World. Nat Bletter, et al. Nov. 1999. SRI International. <http://www.ai.sri.com/~reddy/pubs/vsmm99/>.

[15] Dunn, James, et al. How Much Information? 18 Oct. 2000. School of Information Management, University of California, Berkeley. <http://www.sims.berkeley.edu/research/projects/how-much-info/>.

[16] E-graffiti: evaluating real-world use of a context-aware system. Comp. Jenna Burrell, and Geri Gay. 30 Jan. 2001. Cornell University HCI Group. <http://www.cs.cornell.edu/boom/2001sp/kubo/jburrell_egraffiti.doc>.

[17] Edlund, Stefan, and Jussi Myllymaki. "Location Aggregation from Multiple Sources." Proceedings of the Third International Conference on Mobile Data Management (2002).

[18] Eskin, Hank. Where's George. <http://www.wheresgeorge.com/>.

[19] Extensible Messaging and Presence Protocol (xmpp) Working Group. The Internet Engineering Task Force. <http://www.ietf.org/html.charters/xmpp-charter.html>.

[20] Fischer, Gerhard. "Beyond 'Couch Potatoes': From Consumers to Designers and Active Contributors." First Monday (n.d.). <http://www.cs.colorado.edu/~gerhard/papers/couchpotato-firstmonday.pdf>.

[21] Friendster. <http://www.friendster.com/>.

[22] Galbraith, Megan. Embedded Systems for Computational Garment Design. Thesis. Massachusetts Institute of Technology, 2003.

[23] Ganter, Fritz. Gpsdrive. <http://gpsdrive.kraftvoll.at/>.

[24] Geocaching.com. <http://www.geocaching.com/about/credits.asp>.

[25] Geography Markup Language. Comp. Simon Cox, et al. Vers. 2.0. 20 Feb. 2001. OpenGIS. <http://www.opengis.net/gml/01-029/GML2.html>.

[26] GeoNotes: a real-use study of a public location-aware community system. Petra Fagerberg and Per Persson. Swedish Research Institute for Information Technology. <http://www.sics.se/libindex.html>.

[27] Gimp Toolkit. <http://www.gtk.org/>.

[28] Google. <http://www.google.com/>.

[29] GPS-free positioning in mobile Ad-Hoc networks. Srdan Capkun, Maher Hamdi, and Jean-Pierre Hubaux. Swiss Federal Institute of Technology Lausanne ICA. <http://lcawww.epfl.ch/Publications/Capkun/CapkunHH01a.pdf>.

[30] Hall, Edward T. The Silent Language. N.p.: Anchor, 1973.

[31] iSee. Institute for Applied Autonomy. <http://www.applied-autonomy.com/isee/>.

[32] Ishii, Hiroshi, and Brygg Ullmer. "Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms." Proceedings CHI 97 (1997): 234-241. <http://citeseer.nj.nec.com/ishii97tangible.html>.

[33] Jabber Software Foundation. <http://www.jabber.org/>.

[34] Jabber Studio. <http://www.jabberstudio.org/>.

[35] Jeremijenko, Natalie. Lecture. MIT, Cambridge, MA. 14 Nov. 2001. <http://cat.nyu.edu/natalie/tmtalk/>.

[36] Kaul, Inge. 'What is a public good?" Le Monde diplomatique , June 2000. <http://mondediplo.com/2000/06/15publicgood>.

[37] Kismet Wireless. <http://www.kismetwireless.net/>.

[38] Li, Qiyan. An Architecture for Geographically-Oriented Service Discover on the Internet. Diss. U of Waterloo, 2002. <http://www.shoshin.uwaterloo.ca/publications/pdfs/q2li2002.pdf>.

[39] Lynch, Kevin. The Image of the City. Cambridge: MIT P, 1960.

[40] Millard, Peter G. JEP-0060 Publish-Subscribe. 25 June 2003. Jabber Software Foundation. <http://www.jabber.org/jeps/jep-0060.html>.

[41] Miller, Libby, and Dan Brickley. FOAF: The friend of a friend ' vocabulary. <http://xmlns.com/foaf/0.1/>.

[42] Mitchell, William J. City of Bits. Cambridge: MIT P, 1999.

[43] Mitchell, William J. E-topia: urban life Jim, but not as we know it. Cambridge: MIT P, 1999.

[44] Netcraft. <http://news.netcraft.com/>.

[45] NetStumbler. <http://www.netstumbler.com/>.

[46] "presence." Dictionary.com. <http://dictionary.reference.com/search?q=presence>.

[47] Project Oxygen. MIT Laboratory for Computer Science. <http://oxygen.lcs.mit.edu/>.

[48] Resource Description Framework. <http://www.w3.org/RDF/>.

[49] Salingaros, Nikos. "The Future Of Cities: The Absurdity of Modernism." Planetizen 5 Nov. 2001. <http://www.planetizen.com/oped/item.php?id=35>.

[50] Skyglow. Comp. Steve Albers. NOAA. <http://laps.fsl.noaa.gov/albers/slides/ast/places.html>.

[51] Slashdot Moderation System. <http://slashdot.org/faq/com-mod.shtml#cm520>.

[52] Sutherland, Ivan. "The Ultimate Display." Proceedings of IFIPS Congress 2 (1965): 506-508.

[53] Townsend, Anthony. "Mobile Computing and Communications: New Interactions Between Information Architecture and Infrastructure Use." Position Paper for Bringing Information Technology to Infrastructure (2001). <http://www.informationcity.org/research/icis-workshop/>.

[54] Townsend, Anthony. "The Science of Location: Why the Wireless Development Community Needs Geography, Urban Planning, and Architecture." CHI Wireless Workshop (2001). <http://www.informationcity.org/research/CHIpaper/>.

[55] Using a Steerable Projector and a Camera to Transform Surfaces into Interactive Displays. Claudio Pinhanez . IBM Research. <http://www.research.ibm.com/people/p/pinhanez/publications/chi03a.pdf>.

[56] VanderMeer, Jim. "Hype vs. Reality of Location-Based Services – LBS at the end of 2001." Directions 16 Jan. 2002. <http://www.directionsmag.com/article.php?article_id=144>.

[57] Vogiazou, I T. BuddySpace: Large-Scale Presence for Communities at Work at Play. Inhabiting Virtual Places Workshop at E-CSCW, Helsinki, Finland. Sept. 2003. <http://kmi.open.ac.uk/projects/buddyspace/docs/vogiazou-ecscw03-proposal.pdf>.

[58] War Chalking. <http://www.warchalking.org/>.

[59] Web Feature Server Implementation Specification. Open GIS Consortium. <http://www.opengis.org/info/techno/rfc13info.htm>.

[60] Weiser, Mark. "The Computer for the 21st Century." Scientific American (Sept 1991). <http://www.ubiq.com/hypertext/weiser/SciAmDraft3.html>.

[61] Wikipedia: The Free Encyclopedia. <http://www.wikipedia.org>.

FIGURES

5-6 ‘Reinventing the WDW vacation experience with Destination Disney, ”’ http://www.jimhillmedia.com/articles/12112002.1.htm

6-1 Whale tracking image, http://whale.wheelock.edu/.

6-2 Pileated Woodpecker Count, http://www.nenature.com/PileatedWoodpeckerWinterMap.htm.

6-3 WorldBoard, Jim Spohrer. http://www.worldboard.org/.

6-4 GPSdrive, Fritz Ganter, http://www.kraftvoll.at.

6-6 Cambridge Police, http://www.cambridgepolice.org/images/murdermap.gif.

6-9 Wherify GPS watch, http://www.wherify.com/

7-4 Skyglow study, http://laps.fsl.noaa.gov/albers/slides/ast/places.html.

7-5 Amsterdam Realtime, http://realtime.waag.org/.

7-6 Vert, http://www.vert.net/.

7-7 iSee, http://www.appliedautonomy.com/isee/.

APPENDIX I: EXTENDED USAGE EXAMPLES

The example presented in chapter two, like the client created in chapter four, is a generic device that displays all the information in an area, with some special user interface techniques to narrow the results so that they are not too chaotic and/or out of the user's current context. However, sometimes it is desirable to create single-purpose, domain-specific application clients, so that designers can take liberties within the user interface to tailor the experience to the intended usage. I will outline a few possible specific domains where having access to location data could prove interesting and useful. Consider how each of these improves the legibility of our world through the anchoring of ideas in a spatial context. This section could certainly be quite large as it is relatively easy to think of other applications that combine physical location and information.

Animal Tracking

One such domain-specific client might be an application for creating animal migration and presence maps that could be used by humans to alert them that they are in the migration paths of certain endangered animals such as whales, manatees, or seals. Many of these animals are already tagged with GPS tags and thus have available data streams; the only piece missing is a method to feed this information back to a wider audience. This data retrieval could be integrated into standard navigational displays, such as boater's maps or car navigational systems, so that in addition to showing current location, the display would also present data on any endangered animals seen in the area, thus alerting users to exercise extra caution.

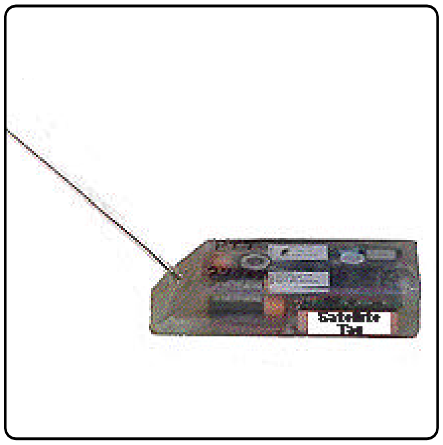


Figure 6-1. Many animals already wear GPS transmitters such as the one above that is attached to whales and tracked via a Website.

A similar application could be designed to allow bird enthusiasts and scientifically charged ornithologists to share their experiences with each other. Each time a bird was viewed, the observer could record his latitude and longitude, as well as the type of bird seen, into his location-enabled application. This information would likely get recorded on his own “home” server (or that of his ISP’s/company’s etc.), which would then be part of the location network that others could search.

Every year the National Audubon Society does something similar to this in their CBC or “Christmas Bird Count,” which encourages people to count the number of birds they see during the course of a single day. In 2001 the CBC netted a total of 54,788,215 birds of all species counted [12]. The data from the CBC is then compared year-to-year in order to study the long-term health and size of the various species of bird populations.

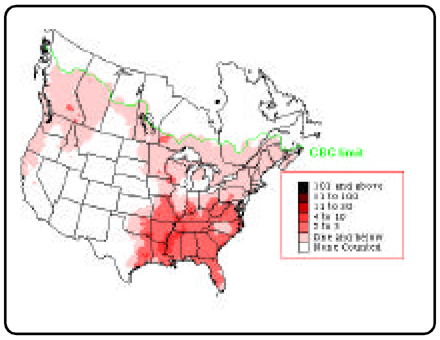


Figure 6-2. Here is a map of one Christmas Bird Count, showing the density of Pileated Woodpeckers found.

Going further in this bird domain, one could imagine that the combination binocular/digital camera recently released onto the market could include the necessary computational power plus an integrated GPS for position sensing. Such a device would be ideal for inclusion of location linked software that could provide access to other users’ findings, as well as supply a portal through which new annotations could be submitted through the push of a button on the binocular.

Urban Infrastructure Viewer

There are numerous other areas that could utilize digital information in a physical context, but still provide a user interface designed for the specific task at hand. For example, the underground infrastructure of the city is a labyrinth of pipes and wires, where often each pipe and wire is stored in a different database. Access to the location network would allow the data to reside in different databases (including non-governmental databases, such as those showing cable routes or Internet connectivity points) and then get reassembled when the need arose. Undoubtedly some of the authentication concerns that necessitate single-sources for this data would need to be replicated in

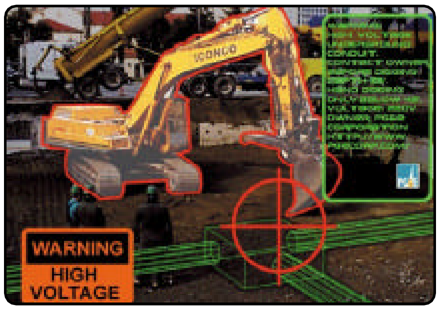


Figure 6-3. Locating utilities could be a specialized application of location linked information systems. Visualized is WorldBoard’s concept of how this might work.

order to reliably use this kind of assemblage of data; however, this is likely possible within the current framework. Intra-company and closed-location networks could be created for such activities where public access is not desired.

The Distributed Wi-Fi Database

Figure 6-4. GPS Drive contains integrated Kismet functionality, but lacks centralized or distributed storage of this data.

A recent entrant into the urban infrastructure is 802.11 or "Wi-Fi" wireless Internet access. Wi-Fi provides high-speed network connectivity for a small area surrounding the network's transmission antenna or "access point." Many of these Wi-Fi networks are configured to allow open access to anyone who wishes to get onto the Internet. Typically, people will purchase network access points for their own office or home and as an ancillary effect will provide free network access to people in their vicinity. Because of this reality it is possible to discover network access in dense metropolitan areas simply by wandering around with a laptop and network detection software, such as Kismet [37] or NetStumbler [45].

Wandering around with a laptop is not always the most efficient or pleasant way to discover areas with network access. Despite the density of hot spots— I took a short drive in the summer of 2002 from MIT to Harvard Square and uncovered 64 of the networks in 2.5 miles— they are somewhat like trash cans: not always there when you need them. To combat human's lack of innate ability to auto-locate network access, people have invented at least two methods for informing the community when they find access. First, centralized Wi-Fi databases accessible via Web sites, and second "war chalking," chalk based symbols scrawled on the sidewalk in areas where network access is available [58]. Along with the community-based efforts, software such as Gpsdrive + Kismet [23] allow you to annotate where you find wireless hotspots for your own future use.

The step from Gpsdrive + Kismet to communal wireless access databases is a small one, but one that would not be available without a proprietary system for linking location with wireless access data. We can generalize this as an ideal-use case: you want to add some location information and share it with others in your community, but you do not want to mess around with the glue layer that bridges the two realms.

Memorial Sites

People do not die online— they live forever through their bits. Unfortunately, we die outside of cyberspace. Sometimes this happens in terrible ways that transform the space for large parts of society, as might be the case for Americans at the Texas School Book Depository where President John F. Kennedy was assassinated. More often though, people die in traffic accidents and their loved ones and the local community need a way to mark the space as a local memorial. On a cross-country trip through the United States, I noticed large numbers of crosses along the highway, anonymous markers of a tragic end. Lately, I have noticed a more personal marker, as mourners tape a picture of the deceased to a tree or signpost near the area where she passed away. This allows a semi-permanent memory/memorial at that location. An extension of the taped picture memorial might be a location nugget placed in the area with links to personal memories and eulogies of the departed.

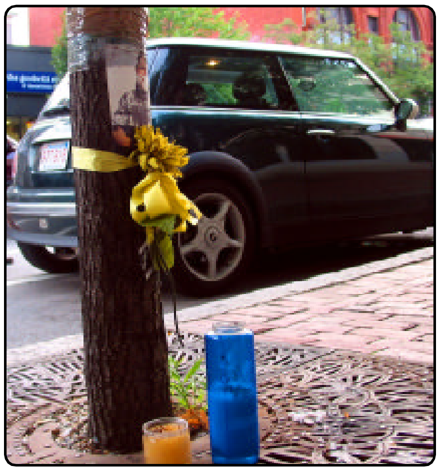


Figure 6-5. Physical memorial sites like this one could be augmented by virtual equivalents.

The Location Linked Information Crime Nugget

It used to be that neighborhoods were close-knit, and news of a crime— no matter how small— would spread house-to-house so that residents drew their own mental maps of areas more dangerous than others. With the rise of sensationalism in the news media, it has become increasingly difficult to surmise how safe an area is from all

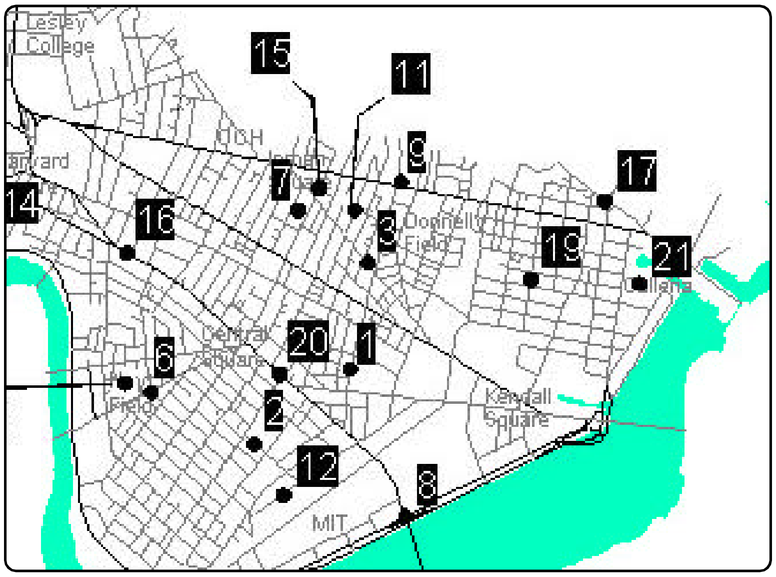


Figure 6-6. This is a "murder map" produced by the city of Cambridge, Massachusetts outlining the number of murders during the 1990s. LLI could be used to create dynamic crime maps for less violent crimes that might be useful to neighbors being aware of their surroundings.

but the most heinous of violent crimes, which are widely reported. Often police departments publish daily crime logs that include everything from the smallest misdemeanor to the "news worthy" felonies. What if these crime logs were keyed to the locations where they actually happened? For example, as a cyclist who has had too many bikes parts pilfered, it would be fabulous to know how many bicycles were stolen from an area so you would know whether you needed to take the seat and tire with you, or if you could save time and keep it on your bike. With a unified location retrieval system, such as is provided by LLI, this kind of community service becomes feasible.

The Geo Roster

The Rendezvous protocol spearheaded by Apple, and now on the IETF standards track, allows devices to find other devices that are on a nearby network segment. Bluetooth provides an alternative protocol implementation of such a location-network coupling. If we extend this concept of "the network coupled with location" further to the reaches of the city, we can start to think about dynamically finding people that are nearby. With this information we can generate a "geo roster" of nearby people. This data could either be entered into your chat roster, such as was done on a different scale in the Buddy Space project [57], or automatically populated in a neighborhood chat room so that everyone in your vicinity could 'hear" any messages you might send. In this way you could talk directly to the neighborhood you might be in, reaching a wider population than you could by standing on a soapbox in the middle of the town square. We could think of this as reviving porch-based communications, but maintaining a degree of anonymity. This pseudo-public speech could be implemented in your geo roster if it was coupled with a Friend of a Friend ('FOAF") [41] data structure to limit people in your nearby chat circle to only those people with whom you have at least one friend in common.

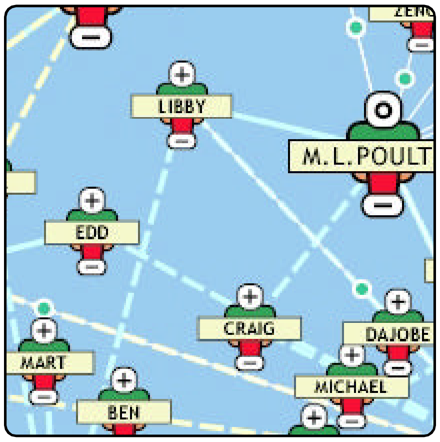


Figure 6-7. Friend of a fried (FOAF) visualization from Foafnaught.

Nearest Resource Locator

How many times have you needed to use the restroom only to be unable to locate the nearest publicly accessible one? Where is your bank's nearest ATM machine? The nearest notary public? These are some of the cases we often hear for location-based services, so rather than rehash their descriptions, I will merely point out that in order for them to be made into a reality they require both location data entry and location data access, both elements in which the proposed system excels.

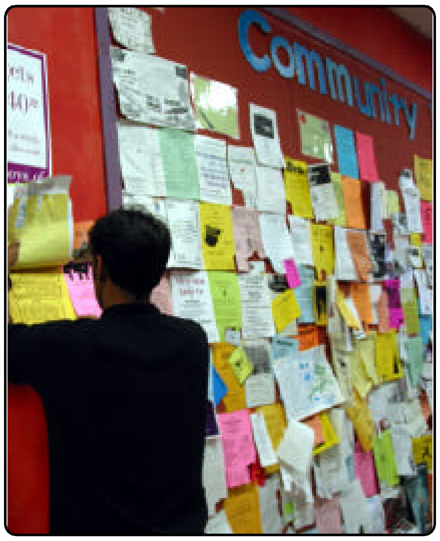


Figure 6-8. Community bulliten boards, such as this one, can help find resources, but is this the best way?

Second-Order Location Applications

The examples given thus far, for the most part, directly utilize the location-based infrastructure to display location-keyed information, usually in the form of a map. One might surmise that second-order applications would emerge that combine location data along with external databases. These second order location applications might analyze the location network to surmise the shortest route between two points based not on direct measurement of the route's length, but on actual travel times.

Zones of Comfort and Location Subscriptions

Figure 6-9. These GPS watches by Wherify, are tied into a closed tracking system, so that parents can see the current location of their children. Similar systems could be created with LLI technologies.

Another second-order application might be an agent that was subscribed to the presence information of various objects, watching for changes to the object's location. These "watcher" agents get notified of changes in the current location of an object, such as a car or a child. The car or child's "owner" would grant watcher agents a subscription to their location presence. Furthermore, these agents can be programmed to be comfortable with an object moving within a certain boundary (such as a neighborhood), but if the watched object moves outside of this comfort zone, an alert would be dispatched to the owner. This sounds eerily close to a Big Brother scenario; however the built-in concept of "subscription" and its functional antidote, the "revocation," explicitly grant the authority for monitoring, and thus individual rights are preserved. Furthermore, by making the subscription process mandatory (not just anyone can watch the locations of these protected types of objects) there is an implicit social contract between the agent and the human (or entity) granting the subscription. This social contract binds the agent to obey her subscription wishes and forbids sharing the location data with other entities not explicitly granted subscription status.

APPENDIX II: COLLABORATIVE MAPS

"Everything is related to everything else, but near things are more related than distant things."
–Tobler's First Law of Geography

In creating LLI, I became interested in emergent systems, in particular maps that grew out of an aggregation of data from a variety of different sources. For a system such as LLI to function, there is a leap of faith that:

1. People will enter data into the system.

2. The data entered will be "good enough" as to be meaningful for those looking at the results.

Where's George

Where's George (http://www.wheresgeorge.com/) is a Website that allows its users to track the movement of their dollar bills as they are circulated throughout the world. Where's George was started by Hank Eskin in 1998 as a curiosity, but has grown into a popular web site in which over $193 million worth of currency has been entered by 1.9 million users [18]. Visitors to the Website enter the serial numbers of their dollar bills as well as their current zip codes, and then when someone else enters the same serial number, the original entrant gets an email notifying him where his bill was found.

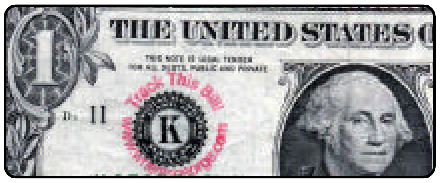
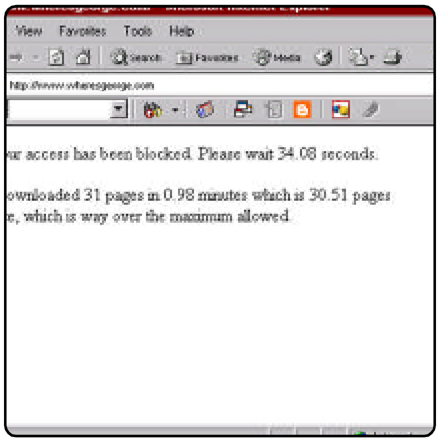


Figure 7-1. "Georgers" stamp currency and then enter them into a web site, tracking their flow through the United States.

I highlight Where's George for a number of reasons, the first of which is the presence of motivation for these millions of people who put information into the site. Where's George, at its essence, is a data structure which would be meaningless without data. The fact that millions of users are willing, without economic compensation, to enter the serial numbers of their dollar bills, with an average likelihood of receiving a "hit" standing at around 4%, is an incredibly interesting phenomenon. Perhaps even more interesting is the way in which the design of the Website cultivates its user base into a community, motivating the users, known as "Georgers" to enter more bills into the system through a competitive ranking, a "George Score," of all users, as well as an active community discussion board, where the social rules of the community are hammered out and judgment on questionable activities is rendered. Where's George is a grassroots movement that explores the interconnectivity of humans and underscores our curiosity to communicate and contribute to a community.

Another point that Where's George raises is the scarcity of computational resources and techniques to counterbalance popularity. While on the one hand the Website is trying to grow its community and serve,as many people as possible, on the other hand it has real computational and bandwidth limitations associated with amassing such an enormous database. With thousands of dollar bills being entered on a daily basis, the database running Where's George has had to limit the number of queries per user in order to fairly serve its community. The site, like other community sites such as Slashdot.org [51], has built in limitations so that if your IP address accesses the Website too often in a given time limit it will block your access.



ur access has been blocked. Please wait 34.08 seconds.

ownloaded 31 pages in 0.98 minutes which is 30.51 pages
e, which is way over the maximum allowed.

Figure 7-2. Where's George blocked access because too many requests per second came from our ip address. Centralized storage requires social behavior to keep serving the public.

Note that Where's George employs a *decentralized data entry/ centralized data storage* design. The decentralization of data *entry* allows the *content* to grow rapidly, from $0 entered in 1997 to $193 million in 2003, however the centralization of data storage causes scarcity of resources and mandated social behavior. Decentralized data storage is probably not the "answer" for this Website either, for it would cause the search and retrieval function to be both inconsistent and less immediate. I do not mean to argue about qualitative merits of the design choices, merely to point it out as an example of the tradeoffs between centralization and decentralization, as illustrated in the diagram below.

The final reason I highlight Where's George is it exemplifies the possibility for secondary, tertiary, and possibly infinite, uses for raw digital data. While the Website was created to track dollar bills as they move from one zip code to another, it generated gigabytes of data that can be mined [15] and reinterpreted for other uses. For example, the Where's George database could be mined to determine the "economic gravity" between any two zip codes in the United States. We could define economic gravity as being the attraction between two micro-economies, that is, two zip codes. How close is 02139 (Cambridge, Massachusetts) and 12603 (Poughkeepsie, New York), or in terms a Georger might use, what is the relationship between the number of hits of bills entered in 02139 and found in 12603 and the national average?

Where's the Population?

Another example application of the Where's George data set, and one that is closer to the main thrust of this thesis, is the assemblage of a population map of the United States made from the zip code hit counts. The hypothesis is that if we were to plot the number of hits a zip code hits, it would be in direct proportion to the population of the zip code, and thus we would be able to *assemble a population map out of millions of points.*

To test this hypothesis I contacted Hank Eskin and he provided me with a database dump of the summary of hits per zip code, which I converted to latitude and longitude coordinates and then plotted. Emerging from the data is a clear map of the United States (figure 7-3), with concentrations of data points surrounding more populous areas. As shown in the image, a strong correlation exists between the Where's George population map and a map created from a NOAA study on light pollution.

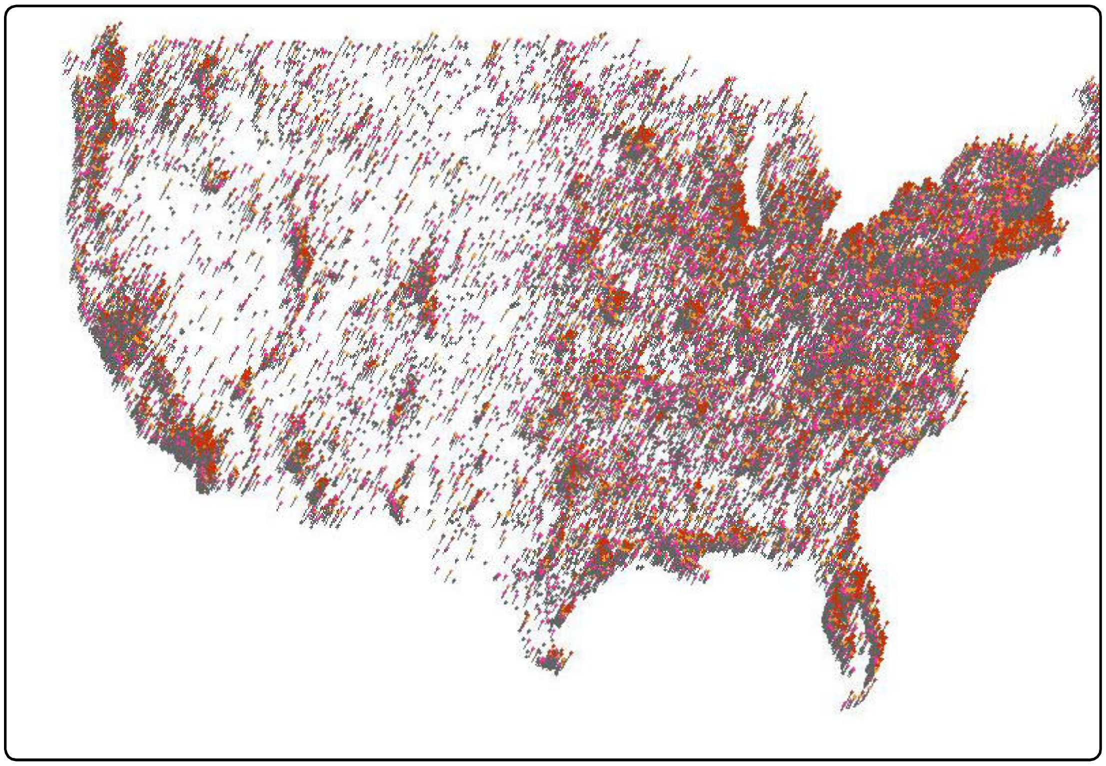


Figure 7-3. Population map of the United States aggregated from Where's George data.

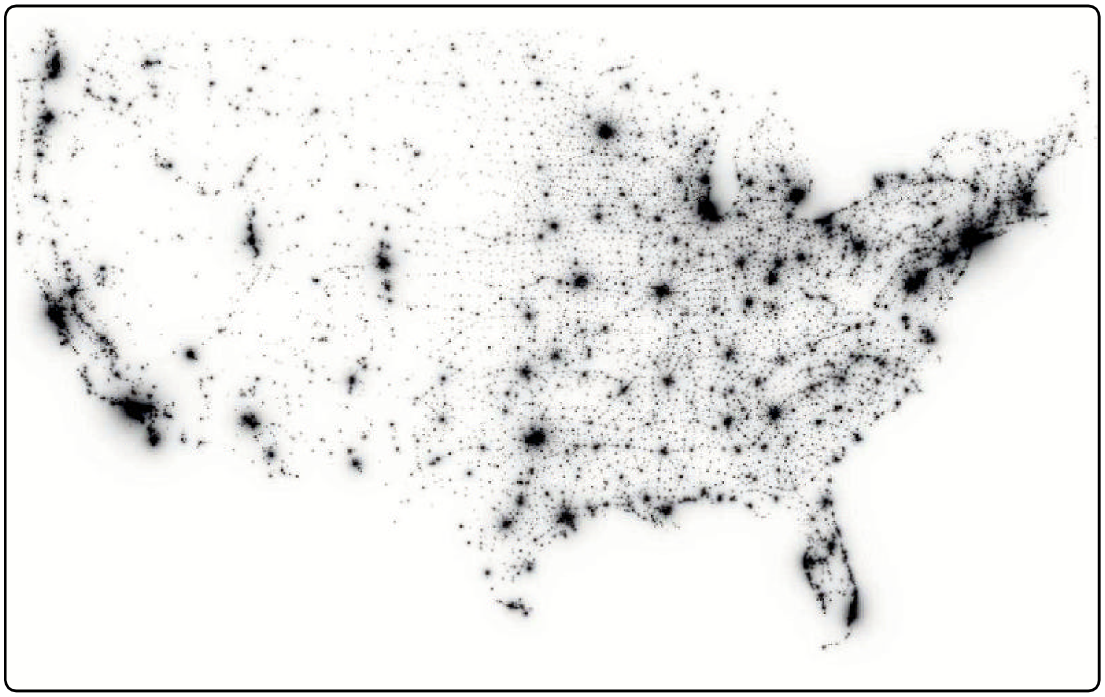


Figure 7-4. NOAA light map of the United States, implying population density.

Amsterdam Real Time

Whereas the Where's George population map was generated out of data collected by millions of users, a project called Amsterdam Real Time gathered location information from seventy-five Amsterdam residents over the course of forty days and still was able to produce an identifiable map with stunning results. Amsterdam Real Time, conceived by the Waag Society for an exhibition entitled "Maps of Amsterdam 1866-2000," displayed dynamic "traces" of the routes that residents take as they navigate the city armed with GPS devices.

When the different types of users draw their lines, it becomes clear to the viewer just how individual the map of Amsterdam can be. A cyclist will produce completely different favorite routes than someone driving a car. The means of transport, the location of home, work or other activities together with the mental map of the particular person determine the traces he leaves. This way an ever-changing, very recent, and very subjective map of Amsterdam will come about. [4]

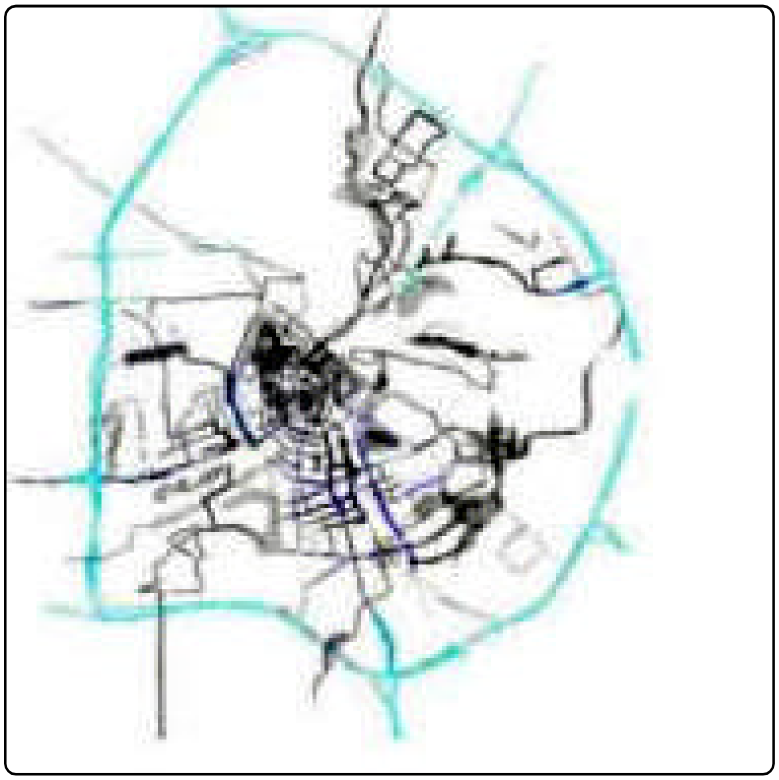


Figure 7-5. Traces made by residents in the Real Time Amsterdam project.

Taxi Traces

A similar venture to Amsterdam Real Time is a map I made from data received from the Boston-based location-based advertising company Vert, Inc. Vert designed a mobile, taxicab-based infrastructure for targeting messages to a particular time and location. While the primary purpose of Vert's business is advertising, they produce residual data that can be graphed to form an always-current map of the city streets of Boston. As Boston has experienced almost a decade's worth of reworking the central artery of the city, where streets might be open one day and closed the next, creating a map of the city is an exercise in futility. However a recognizable image (the plan of Boston's roads) emerges from position broadcasts of taxicabs during their normal routes throughout Boston. Whereas there were millions of dollars entered in the Where's George map, seventy five people in Amsterdam Real Time, there were less than ten taxis used to create the taxi traces.

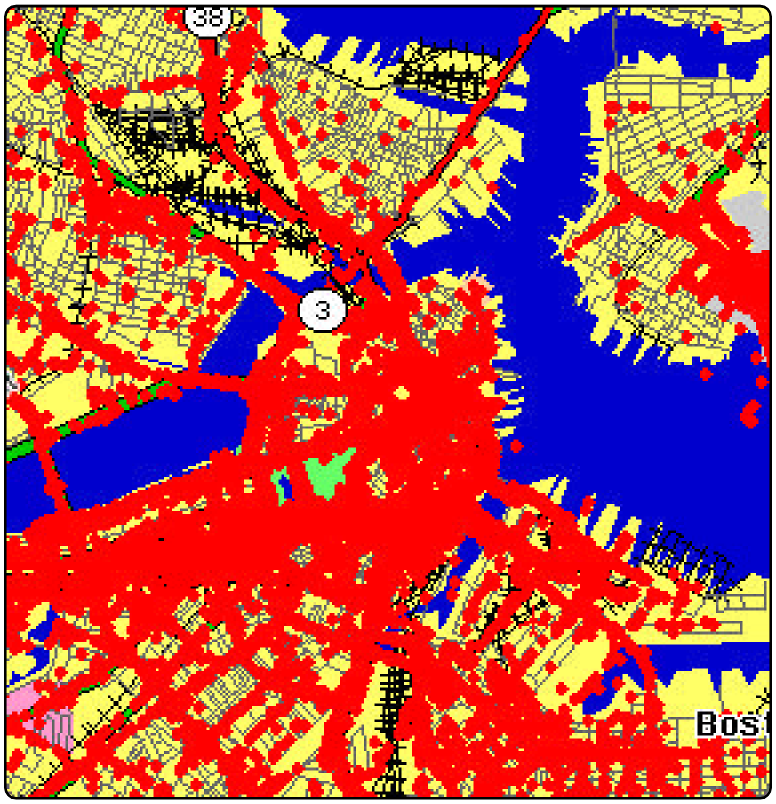


Figure 7-6. Vert's taxi traces, plotted as red dots overtop a map of Boston.

NYC Surveillance Camera Project/iSee

The Institute for Applied Autonomy has created a Web-based application called "iSee" [31] that will plot the "route of least surveillance" between two points in New York City. iSee uses a database of closed-circuit camera positions as gathered from members of the public at large and entered into the NYC Surveillance Camera Project website [63]. There are a few of things to be pointed out about this project: first, the NYC Surveillance Camera Project is a grassroots movement to document the encroachment of technology and utilizes the distributed resources of concerned citizens as its data-gathering device. Anyone can add a new camera location, which is both a strength and a weakness. Secondly, iSee was authored on top of the NYC Surveillance database.

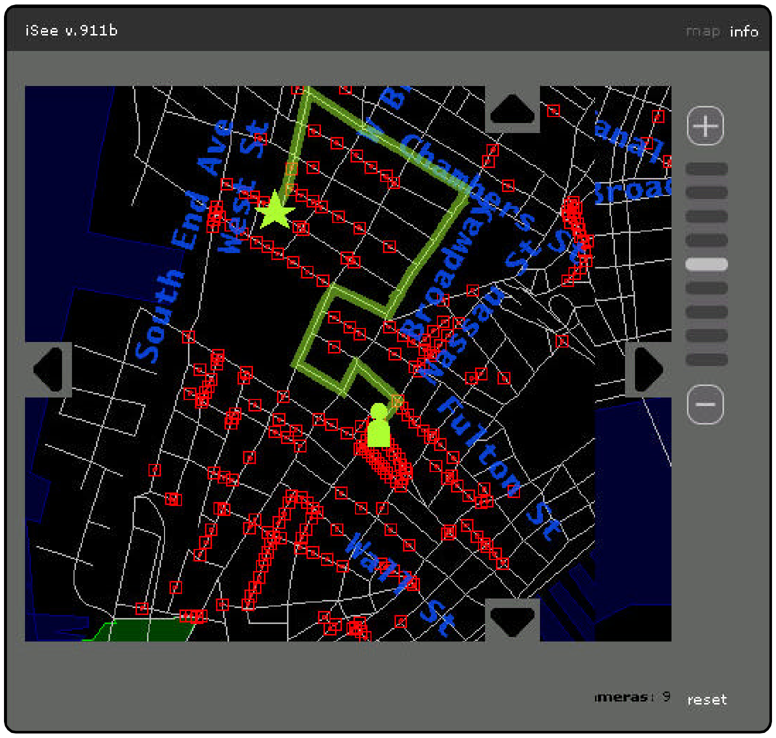


Figure 7-7. The iSee project shows the path of least surveillance between two points in NYC.